



**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**  
(Sprint Docket No. 1731)

In re Application of: )  
Kevin Packingham )  
Serial No.: 09/963,776 ) Group Art Unit: 2655  
Filed: September 26, 2001 ) Examiner: Matthew J. Sked  
For: Method and System for Use of Navigation )  
History in a Voice Command Platform )

Mailstop Amendment  
Commissioner for Patents  
PO Box 1450  
Alexandria, Virginia 22313-1450

**DECLARATION PURSUANT TO 37 C.F.R. SECTION 1.131**

Dear Sir:

I, Kevin Packingham, hereby declare:

1. I am the first, original and sole inventor of the subject matter claimed in United States Patent Application Serial No. 09/963,776, filed on September 26, 2001, and entitled, "System Method and System for Use of Navigation History in a Voice Command Platform." I submit this Declaration to overcome 35 U.S.C. § 102(e) and 35 U.S.C. § 103(a) rejections of my patent application based on United States Patent Publication No. 2004/0179659 to Byrne et al. (hereafter "Byrne"), filed August 21, 2001.

2. I am now and have been at all relevant times described herein, an employee of Sprint Spectrum L.P. (hereafter "Sprint") working in Sprint's research and development facilities in Overland Park, Kansas.

3. The inventions disclosed in the above-captioned patent application were conceived by me well prior to August 21, 2001. I also exercised due diligence in reducing these inventions to practice from at least just prior to August 21, 2001 until the invention was constructively reduced to practice as evidenced by the filing of the present application on September 26, 2001.

4. Accompanying this Declaration are copies of various Exhibits 1 – 8. These Exhibits support the conception of my invention prior to August 21, 2001 as well as my efforts to diligently reduce my inventions to practice from at least just prior to August 21, 2001 until the constructive reduction to practice of my inventions.

5. Attached as **Exhibit 1** is an email message that was sent from Mr. Steven Funk, in-house patent counsel for Sprint, to Mr. Lawrence Aaronson, outside patent counsel for Sprint. In this email, which was sent well prior to August 21, 2001, Mr. Funk refers to an attached document that I authored also well prior to August 21, 2001. This document, which describes Sprint's requirements for voice command platform features, is attached here as **Exhibit 2**. The dates in Exhibits 1 and 2 have been redacted as is permitted by MPEP § 715.07.

6. As is described in Exhibit 2, I conceived of implementing, as part of a voice command platform, a method and system for transitioning a user of a voice command platform to the use of an expert mode prompt set based on navigational patterns of the user. (*See* page 7, paragraph 8.4). Consequently this document evidences my early conception of a method and system for expert-mode transition as is claimed, for example, in claims 12, 13, 20-22, 24-31 as were filed in the present application. A listing of the claims as filed in the present application is attached here as **Exhibit 3**.

7. As is also described in Exhibit 2, I further conceived of implementing, also as part of a voice command platform, a method and system for managing user disconnects (e.g., due to hang ups, signal loss, etc.) from such platforms. The method and system I conceived of included maintaining a navigational history for users of the voice command platform to allow such users to resume a voice command session at the point of a disconnect from the voice command platform. (*See* page 8, paragraph 18). Consequently, this document also evidences my early conception of a method and system for managing system disconnects as is claimed, for example, in claims 1-11 and 15-19, as were filed in the present application. (*See* Exhibit 3).

8. Attached as **Exhibit 4** is an email I received from Mr. Aaronson on July 31, 2001, well prior to the filing date of Byrne. As indicated in this email, Mr. Aaronson had attached to the email a draft consolidated written description that formed the basis of four separate but related patent applications, including the present application (designated Sprint Docket 1731). The consolidated written description referred to in Mr. Aaronson's email is attached here as **Exhibit 5**. This document, which is dated July 30, 2001, further demonstrates the conception of my inventions prior to August 21, 2001. Embodiments of the inventions that I conceived of and that are claimed in the present application are described in this document, along with the subject matter of the three other related patent applications that Mr. Aaronson was preparing as a group. In fact, this document clearly supports each of the claims as filed in the present application in accordance with 35 U.S.C. § 112. For example, an embodiment of a system in accordance with claim 1 is described on page 14, lines 1 – 9 and on page 20, lines 1 - 8 of Exhibit 5. The navigation-recording logic features of claims 2 and 3 are described on page 28, lines 1-17. The session-restore features of claims 5-8 are described on page 41, line 13 through page 42, line 21. Claims 9 – 11 are directed to additional aspects of maintaining a navigation history and session

restoration. These claims are supported by at least the sections of the application cited above. The expert mode-transition aspects of claims 12 – 14 are described on page 38, line 3 through page 41, line 10. The wireless link aspect of claim 14 (as well as the wireless link aspect of claim 23) is supported on page 41, lines 13-17 of Exhibit 5. Claims 15 – 23 are directed to a method implemented in a voice command platform, such as the voice command platform of claim 1, and contain similar limitations as claims 1 – 14. Thus, claims 15-23 are similarly supported by the disclosure. Claims 24 – 27 are directed to a voice command platform including logic for switching the use-level of a user, such as to an expert mode. These claims are supported, at least, by the sections of Exhibit 5 cited above, such as on page 38, line 3 through page 41, line 10. Claims 28 -35 are directed to a voice browser system for executing voice-tag applications. These claims are supported, at least on page 15, lines 5-15; on page 34, line 16 through page 35, line 2; and by the sections cited above.

9. Attached as **Exhibit 6** is an email message that I received from Mr. Aaronson on August 7, 2001, also well prior to the filing date of Byrne. As indicated in this email, Mr. Aaronson had attached, for review and comment, four draft sets of claims, including a set of claims for the present application. The draft claim set for the present application referred to in Mr. Aaronson's email is attached here as **Exhibit 7**. A comparison of this draft set of claims with the claims of the application as filed (Exhibit 3) demonstrates that the claims in Exhibit 7 are identical to those filed with the application, with the following exceptions: (i) claims 14 and 23 as filed in the application were added to specify a wireless link and (ii) claims 24 and 25, as filed in the application, were originally a single claim. Consequently, this draft claim set also further evidences the conception of my inventions prior to August 21, 2001. Furthermore, this

communication from Mr. Aaronson also demonstrates diligence in reducing my inventions to practice from just prior to August 21, 2001.

10. Over the next three to four weeks following Mr. Aaronson's email of August 7, 2001, my colleagues and I were busy reviewing the four draft claim sets including the claims for the present application (Exhibit 7) and the consolidated written description (Exhibit 5) for the four related applications. After our review, we provided feedback to Mr. Aaronson regarding the draft claims and the consolidated written description.

11. Attached as **Exhibit 8** is an email thread of messages between Mr. Aaronson and me regarding preparation of the four related applications, including the present application, that were referred to above. As further evidenced by this email thread, Mr. Aaronson and I were diligent in preparing the present application and, thus, constructively reducing my inventions to practice. Specifically, the first email in this thread (on page 2) was sent on August 28, 2001. This email was an inquiry from Mr. Aaronson to me regarding inventorship of the inventions for the four applications sharing the consolidated written description of Exhibit 5, including the present application. The second email of this thread (on the bottom of page 1) was sent on September 2, 2001. This email was my reply to Mr. Aaronson's inquiry regarding inventorship. In this email, I informed Mr. Aaronson that I am the sole inventor of the inventions claimed in the present application. The third email in this thread (on the top of page 1) was from Mr. Aaronson to me. In this third email, Mr. Aaronson indicated that the four final patent applications, including the present application, and associated formal papers were sent to me from Mr. Aaronson's firm on September 5, 2001.

12. The inventions claimed in the present application were then constructively reduced to practice on September 26, 2001, as is evidenced by the filing of the present

application. During the period between September 5, 2001 and the September 26, 2001 filing date of the present application, my colleagues and I were busy reviewing and executing the four related patent applications that were prepared by Mr. Aaronson and share the consolidated written description included here as Exhibit 5.

13. All of my inventions disclosed in the present application were conceived of and reduced to practice in the United States.

14. Various dates have been redacted from the attached Exhibits. However, all of these redacted dates are on or before August 21, 2001, the filing date of Byrne.

15. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: 7-11-2005



Kevin Packingham

-----Original Message-----

From: Funk, Steven J.  
Sent: [REDACTED]  
To: 'aaronson@mbhb.com'  
Cc: Nehls, Timothy A.  
Subject: New Sprint Docket 1698

I have had several conversations with Kevin Packingham (inventor) regarding Sprint's development work for our Voice Command system. There are several concepts which Sprint has worked on and/or came up with the concept for our vendors to incorporate into the Voice Command system. IBM is the main vendor and integrator of the technology. Nuance Communications, Converse Network Systems and Telcordia are working with IBM and Sprint on this system. The Voice Command system is being enhanced to include a voice browser and other new features.

The attached Word document discusses the Sprint requirements. The concepts which are not known and not available from the above vendors or other third parties, to the best of our knowledge, would include the following (the numbers refer to the paragraph numbers in the requirements document):

- 3. System Level Grammars
- 8.1 Personalized Grammers
- 8.2 Notification Profiles
- 8.3 Personalized Prompts
- 8.4 Expert Mode Transition
- 14. Secondary Dictionary
- 18. System Disconnects
- 20. Bookmarks
- 26. Text to Speech.

If his schedule permits, I will arrange for you to meet with Kevin on June 20 in KC.

From a strategic standpoint, lets discuss how we want to cover these various items (One big application with possible subsequent continuation applications, or several distinct applications). Please call me at your convenience after you have reviewed the material.

Thanks.

Steven J. Funk  
Sprint Law Department (Intellectual Property)

913-624-6828 voice  
913-624-6388 fax  
steven.j.funk@mail.sprint.com

-----Original Message-----

From: Packingham, Kevin D. PCS.  
Sent: [REDACTED]  
To: Funk, Steven J.  
Cc: Packingham, Kevin D. PCS.  
Subject: VC Architecture

Steven,

This is a nice presentation on the architecture of the VC platform...

<<SN-VAD brief 000908.ppt>>

This is the current design specification for our future implementation of the VC Browser. Let me know if you think we should move forward with a patent application.

<<Voice Browser UER 1\_3.doc>>

Kevin Packingham  
SPCS Product Design and Usability  
Office: (816) 559-1076  
PCS: (913) 485-6508  
Fax: (816) 559-3400



**Sprint PCS<sup>SM</sup>**  
Sprint Personal Communication Services<sup>SM</sup>

## **User Experience Requirements**

From: Kevin Packingham	Issue Date: [REDACTED]
Title: UER for Voice Browser Integration with Voice Command	

### **1. Introduction**

The Voice Command (VC) product was originally designed with the intention of expanding the applications supported by the platform. The first application that was launched with the platform was Voice Activated Dialing (VAD). The VAD application gives customers the ability to dial numbers in a personal address book by speaking a name.

The Voice Command application suite will now be enhanced to include a Voice Browser (VB). The VB will allow customers to listen to content such as news, sports, stock quotes, etc and navigate to a wide variety voice web applications hosted by external content providers. The enhanced design for VC is predicated on the goal of maintaining existing functionality and insuring that the ability to voice dial is not complicated by the new applications.

This document is intended to give an overview of User Experience that must be maintained during the development of this product. Detailed dialog design will be provided after the Statement of Work has been completed and the scope of the project is clearly defined. Temporary prompt sets will be provided during development. Final recordings will be provided prior to STIC delivery.

### **2. Overall Architecture**

With the inclusion of additional applications it is necessary to modify the basic organization of the VC interface. The VB must be fully integrated into the call-flow as the user begins his/her interaction with the system and launch during the call setup and initiation. Therefore, the VB must be responsible for transitioning the user between applications, including VAD. This design will allow the user to create personal profiles and potentially modify the opening prompts. This implementation will most likely necessitate the conversion of the VAD application so that it complies with the Voice XML 1.0 specifications.

The VB platform must support full integration of all SPCS and Sprint applications. This includes all voice-activated aspects of the unified communications and voicemail applications. The VC platform must be the centralized integration point for all voice recognition applications on the Sprint network. It must also include a secure connection to SPCS data facilities and Sprint e|Solutions hosting facilities.

### **3. System Level Grammars**

The VB must have an integrated Root Application, or equivalent process, that maintains a global grammar set for the entire VC application. The VC Root Application must be active without being called or identified in any Voice XML document. Each content provider must be able to specify a separate root application that is independent of VC process. No more than one root application may be specified in each document based on the standard.

The global VC Root Application must support the following grammar:

Navigation	Grammar
Initial Ready Prompt	Voice Command   Voice Dialing
Full Content Menu	Main Menu
Previous Menu	Stop   Cancel
Pause	Pause   Go to Sleep
Save	Bookmark   Save
VC Help	Voice Command Help

This grammar set must be active throughout any VB session. If the content provider creates a document or root application that clashes with the VC Root Application then VC must always take precedence. DTMF will be used to reinitiate the browser after a Pause or Go to Sleep.

## 4. Default Behavior Following \* Talk

### 4.1. Prompts

The VC application currently plays a "Ready" prompt and waits for the user to speak a command. This behavior will be unchanged for the standard user who has not created a personal profile. The system must play the "Ready" prompt and then pause for a VAD command. After pausing for approximately 4 seconds the system will play a series of prompts that gives the user some examples of content areas that can be accessed.

**System:** tone, *Ready*

**User:** silence

**System:** *To listen to web content say the content area such as News, Sports, Email, Voicemail, Stock Quotes. To hear a complete list say What are my options.*

Following the prompts with content examples the system should pause for an additional 4 seconds. After the pause, or if the user speaks "What are my options", the system should play a complete menu of the content. Once the user has heard the menu the "Ready" prompt should be repeated followed by another pause. If the user does not speak a command the system should timeout, give the silence error, and play the full menu again.

**System:** tone, *Ready*

**User:** silence

**System:** *To listen to web content say the content area such as News, Sports, Email, Voicemail, Stock Quotes. To hear a complete list say What are my options.*

**User:** silence

**System:** *Your options are News, Sports, Stock Quotes, Horoscopes, Lotteries, Traffic, Driving Directions, or Sprint Express.*

**User:** silence

**System:** *I didn't hear anything, previous prompt*

### 4.2. Grammars

The VAD grammars must be active from the "Ready" prompt. This includes all personal and dialing list address book entries. The scope of the VAD grammar should not extend beyond the top level.

The default grammar must also include all content areas. The grammar for the core content areas must be scoped globally and be active throughout the VB navigation. The core grammar may include specific content providers in two instances: 1) a negotiated placement based on business development criteria and 2) user established personal preferences. If the user chooses to create a personal profile then the content providers that were identified in that profile must be

active in the default grammar. Since the core grammar may change periodically it must be configurable from an administrative interface. The user must have the ability to retrieve a full list of menu options for the voice and web interfaces.

#### 4.2.1. Level One Grammar

Navigation	Grammar
Initial Ready Prompts	Start over   (Go to) Beginning
Full Content Menu	What are my [options, choices]
Content Selection	That one
Previous Menu	Back   Go Back   Previous
VC Help	Help   Voice Command Help

#### 4.2.2. Negotiated Grammar Placement

No contractual placement in the Voice Grammar will be implemented without prior approval from Product Design and Usability. There are a variety of restrictions that will be developed as the product evolves that will guide the Business Development organizations as new agreements are negotiated. In general, each top-level keyword (grammar) must be evaluated in terms of its potential to clash with other Voice Command grammars, produce recognition strength, and enhance the customer experience with the product. During the negotiation of business terms with each content provider the appropriate keywords should be reviewed.

### 5. Level Two

#### 5.1. *Prompts*

The prompts for all second level applications will be customized based on business agreements with specific content providers. The format for the prompts must follow the following format:

(Category) from (Provider), or say What are my options for more choices

For example, "Driving directions from Mapquest, or say What are my options for more choices". If the user does not barge-in the system should begin playing the content. If the user has created a personal profile then the content from the default content provider should use the profile when appropriate.

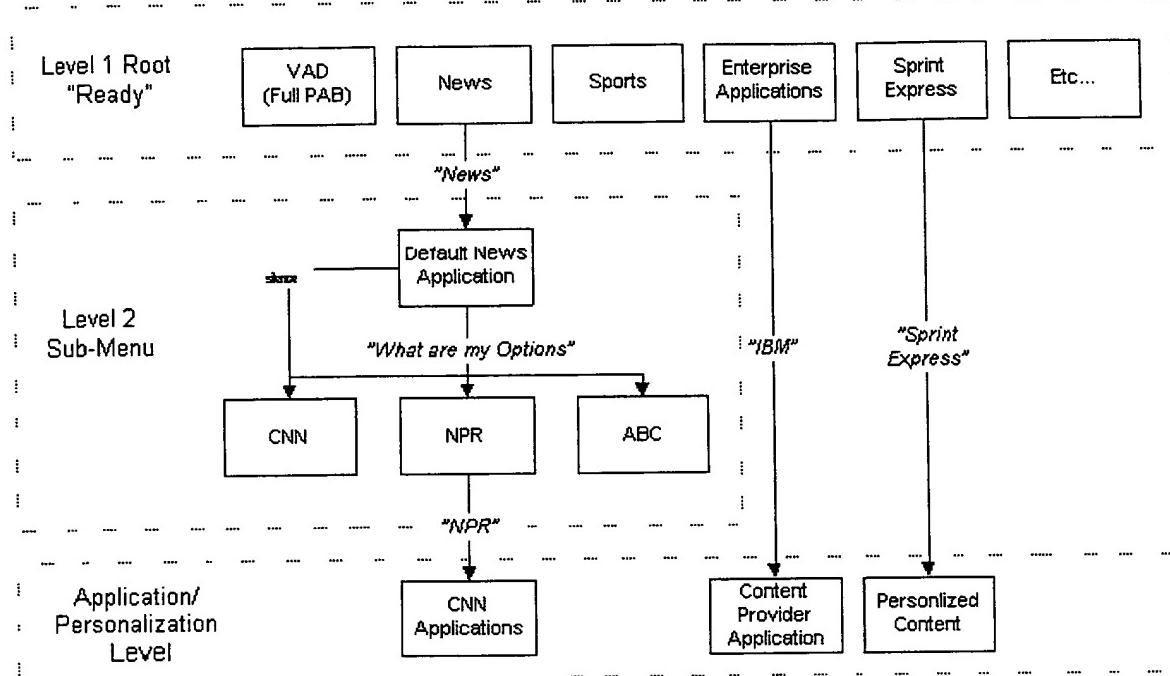
#### 5.2. *Grammars*

The grammar for each content category will be content specific. The grammar must include all content providers and the key navigation elements associated with that content.

Navigation	Grammar
Initial Ready Prompts	Start over   (Go to) Beginning
Full Content Menu	What are my [options, choices]
Next Content Provider	Next   Forward
Previous Menu	Back   Go Back   Previous
Profile Update	Add to (my) profile
Content Selection	That one
VC Help	Help   Voice Command Help
All major content categories	Go to ~

## 6. General Architecture

The levels of the VB will consist of the top two levels described in the previous sections and an application level. IBM must satisfy the requirements for the top two levels and the integration of Bookmarks and Sprint Express applications.



## 7. Within Application/Content Navigation

### 7.1. Prompts

The content provider will determine the prompts for each application. All Premium Placement content providers must work with SPCS to develop the appropriate dialog and grammar design. All content providers that are listed with the menu structure must comply with the design standards established by the SPCS Voice Interface Style Guide.

### 7.2. Grammars

The grammar for each application will be content specific. The following Voice Command grammars should be active within each application.

Navigation	Grammar
Start of Application	Start over   (Go to) Beginning
Full Content Menu	What are my [options, choices]
Forward within Content	Next   Forward
Previous Menu	Back   Go Back   Previous
Profile Update	Add to (my) profile

## 8. Personalization

Each user must have the ability to personalize his/her experience with the VB through the web site and voice interface. The personal profile must be shared with all SPCS applications and products

(i.e., Wireless Web). The user must have the ability to specify sets of profile data that can be shared with content providers.

The fields for personalization must include, but are not limited to:

- Stock Quotes
  - Up to 3 stock portfolios
  - Text entry fields for 12 stock ticker symbols each
  - Name field for each account
- Email address for notifications
- Permission to use Location Based Services (i.e., driving directions)
- Driving Directions
  - Full home address (street1, street2, city, state, zip)
  - Full work address
- Messaging profile
  - Notification and prompt options
  - Voicemail, Wireless Web Mail, and Email Reader configuration
- Email Reader
  - Up to 3 accounts for email reader (email provider, protocol, username, password, reply address, name)
  - Name field for each account
  - Specify default playback mode (subject line, body)
  - Playback rate (standard, fast)
- Corporate Email Reader (Lotus, Exchange, etc.) – same as standard email reader
- Joke of the day – category preferences
- Natural language search preferences
- Conference Calling
  - Up to 5 profiles with labels
  - 10 names and phone numbers per profile
- E-Wallet – (credit card information)
  - First name
  - Last name
  - Middle Initial
  - Billing Address
  - Shipping Address 1
  - Shipping Address 2
  - Credit card type Credit card number
  - Expiration Date
  - Password (possibly for access)
  - Extra fields
- Financial Manager
  - Up to 5 accounts
  - General access number, account number, PIN
- Traffic routes (state, city, road)
  - Up to 10 frequently traveled roads with persistent ordering
  - Option to reverse order of playback by am/pm
- Weather – Up to 10 cities (country, state, city) with persistent ordering
- News
  - Up to 10 news sources with persistent ordering
  - Up to 10 topic areas (categories) of interest for each source
  - Report type for each
- Sports
  - Up to 10 sports (sport, league, team) with persistent ordering

- Report type for each
- TV Dramas – Up to 10 with persistent ordering
- Reminders (birthdays, home/auto maintenance, etc)
  - Up to 50
  - Label, date, and reminder frequency
- Entertainment (television, music) - Up to 10 with persistent ordering
- Gaming profile
- Lottery (state, lottery) – Up to 10 with persistent ordering
- Horoscopes – Up to 12 with persistent ordering
- Bookmarks – Up to 10 with persistent ordering
- Portal / Web Favorites (AOL, Yahoo, etc)
  - Up to 10 with persistent ordering
  - Username and password
  - Report type for each
- Internet Radio – Up to 10 stations
- Voicemail
  - Up to 3 with persistent ordering
  - General access number, account number, PIN
- Restaurants – Up to 10 categories
- Password Vault
- Movies
  - Up to 3 theaters
  - Report type for each

### **8.1. Personalized Grammars**

Users must have the ability to create a custom set of grammars for the major content categories using the VC web site. When the user creates a set of custom grammars the system must evaluate the inclusion of the grammar and prevent the user from creating clash scenarios. The personalized grammars must be included in parallel with the default grammar. For example, a user may choose to access the stock quote application by saying “check my portfolio”. This functionality will be separate from the Bookmarks application.

### **8.2. Notification Profiles**

Users must have the ability to get a summary of their Unified Communications account and any items identified in the Reminders application. After the user connects to the system they should hear the summary prior to the Ready prompt. The user must be able to retrieve the summary as a Level 1-menu option. The inclusion of the message summary with the Ready prompt will be controlled through the web site. The retrieval of the message summary must not take more than 5 seconds.

All status information associated with notification profiles must be contained in a data push to the MS-SMS on the VC platform. The data must be in XML format and must only contain VC subscribers. Up to 10 fields must be available for applications that permit user notification. The MS-SMS must distribute the data to the appropriate SN.

### **8.3. Personalized Prompts**

Users must have the ability to choose from a set of personas. All personas will be created, approved, and maintained by SPCS. No more than 10 prompt sets will be maintained for American English. Foreign language prompt sets will be activated using the same process but will not be counted as part of the core 10 prompt sets. A placeholder must be included for Spanish during the initial development. Users must be able to switch prompt sets from the web or the voice interface. Each interface must include a sample of the prompts. All prompt sets must include both standard and expert mode recordings.

#### **8.4. Expert Mode Transition**

The system must monitor the navigational patterns of the user. SPCS will determine a threshold at which the user will be prompted to transition to an expert mode prompt set. For example, after accessing the VB on 30 separate occasions the user will be prompted with the benefits of expert mode and given the option of transitioning to expert mode. The user will be prompted to transition to Expert Mode only once.

#### **8.5. Application Initiation**

Usage must be tracked on a per application basis. If a user is interacting with the application for the first time the system must provide detailed instructional prompts regardless of experience level with the VC platform (i.e., expert vs. novice). Once the user has completed the initial introduction to the application the system should use the profile setting for experience to determine the prompt set.

#### **8.6. Web Site Design**

The Voice Command web site must be modified to include the personalization options. A minimum of two additional links will be added to the left navigation. The design of the pages will be determined by SPCS.

#### **8.7. Sharing Customer Profile Data**

The VC application must include a pre-determined procedure for passing subscriber level information to content providers. The data set must not include MIN or Social Security Number as identifiers. The data set must include Subscriber ID and any relevant personalization fields. The exact mechanism for passing the profile data will be determined during development. It is anticipated that an executable procedure will be developed. The procedure must be accessible to content providers with a predefined set of variable names. The data transmission method for providing personalization profiles to content providers must not break the application if the content provider is not capable of using the data.

The process for retrieving customer profiles should be consistent with the following procedure:

1. VB session is initiated with subscriber ANI being passed to server
2. Server creates a session and pulls all of the customer profile information from the database
3. Create a sub-dialog for retrieving variables

Data should be shared between applications. For example, the personal address book must be accessible by the voice dialer and the email applications.

### **9. Audio Status**

When the system is processing data background audio must be played to indicate system status. All delays longer than 2 seconds must include the fetch audio and clearly indicate that the system is processing information. All prompts for "Please continue to hold" must be replaced with the audio status prompt.

### **10. Grammar Format**

The VB must support both JSGF (Java Speech Grammar Format) and XML grammar formats.

### **11. Administrative Interface**

All configuration of the VB must be accessible through a web based administrative interface. The administrative interface must only be available to ESP and existing System Administrators. This

functionality would be another option of the existing VC System Administrator interface. The interface must include the functionality necessary to:

1. Update Global, Level 1, and Level 2 grammars
2. Upload new prompts
3. Modify the prompt options available to subscribers
4. Identify content providers as a Premium Service
  - Identify price point for per usage service – options must include \$.25; \$.50, \$1.00, \$1.50, \$2.00, \$2.50, \$3.00, \$3.50, \$4.00, & \$5.00
  - Identify MRC plans that should have access
5. View and modify the subscriber profile
6. Add personalization data fields to dB and web-site
7. Number of connections before prompting for Expert Mode transition
8. Thresholds for initiating Disambiguation
9. Timeout for browser history following disconnects and hang-ups
10. DTMF mapping for content providers and the corresponding URL

## **12. Enterprise Solutions**

The existing Group Administrator interface must be updated to include the option of adding Dialing List users to the Enterprise Voice Web Application profile. Group Administrators must have the ability customize the menu structure of the corporate users who have access to the VB. Once the user has been added to the Voice Web the menu structure of the user must reflect the structure established by the Enterprise.

## **13. DTMF Backup**

Users must have the ability make selections using DTMF. The DTMF option must be on by default for all prompts. The prompts for DTMF options should only be presented after the 3<sup>rd</sup> consecutive mis-recognition. All applications must have a default DTMF mapping based on the first three characters of the content providers menu label (i.e., CNN = 266). The prompts following the 3<sup>rd</sup> mis-recognition must give users the option of speaking the name or using the keypad to enter the first three letters. If there is more than one content provider that maps to the 3 digit code the system must disambiguate using sequential digits (i.e., "For CNN press 1, etc..."). Any content provider or application that has an existing interface that includes DTMF should maintain a consistent mapping (i.e., Voicemail).

## **14. Secondary Dictionary**

SPCS will control the master dictionary for the VB. Content providers must have the ability to specify secondary dictionaries for context specific grammars. For example, content providers that specialize in driving directions will need to update the dictionary with cities and street names. SPCS will not update the core dictionary for individual content providers.

## **15. Disambiguation**

The VB must have the ability to disambiguate between navigational elements when the confidence level is below a specified threshold. This functionality must include three recognition categories: a) High confidence – proceed without confirmation, b) Moderate confidence – disambiguate or confirm, and c) Low confidence - reprompt. The thresholds for disambiguation must be configurable from the System Administrator interface.

## **16. Security**

The VB will access content through the IT Gateway. The connection between the VB and the IT Gateway will be dedicated and secure. The transmission of data will follow standard Internet Protocols. SSL must be supported between the IT Gateway and the content provider.

## 17. Out-Dialed Content Providers

It must be possible to initiate a call to a content provider in certain circumstances. However, once the user has out-dialed a call from Voice Command they will be not be able to re-enter the voice browser without terminating the call and dialing \* Talk. All out-dialed calls must be to toll-free phone numbers.

The prompts for all out-dialed calls will be consistent throughout. The prompt must clearly indicate to the customer that they are exiting the Voice Command application.

**System:** tone, Ready

**User:** Flowers

**System:** Placing a call to FTD Flowers, or say what are my options.

**User:** silence

**System:** When you are finished hang up and dial \* Talk to return to Voice Command.

## 18. System Disconnects

If the user hangs up or is disconnected from Voice Command the system must retain his/her navigational location within the application for 15 minutes. The length of the history persistence must be a configurable parameter from the System Administrator interface. The prompts following a VAD call should always default to the standard Ready prompt. Disconnects include coverage drops, signal fade, hang-ups, and out-dialed calls. Users must be prompted with the appropriate information to regain orientation within the application. The prompt must include the ability to resume from the previous point or start over from the Ready prompt. For example, if the user is in step 5 of a driving direction when they lose coverage the system must prompt the user with "Step 5 of the driving directions from X to Y, or say Voice Command to start over".

## 19. Sprint Express

The VB must include the ability to create a personal profile that includes a list of content categories and preferences within those categories. The user will then have the ability to hear the updated content based on his/her preferences. Users must have the ability to control the content category, order of presentation, content provider, and content details. For example, a user may want to hear CNN News followed by traffic in Dallas. Users must be able to move to the next category with the "Next" keyword. The Sprint Express will be an application that resides at the same level as the other major content categories and the user will need to select the Sprint Express option before the automatic content playback is initiated.

### 19.1. Prompts

The prompts for the Sprint Express will be identified during development. The Sprint Express application must have a unique identification. Users must have the ability to create a profile if they navigate to the Sprint Express without having created a profile on the web. If the user has already created a profile the system should begin playing content immediately.

**User:** Sprint Express

**System:** Sprint Express gives you the ability to personalize your content. You do not have a Sprint Express Profile. Say the name of the content category you would like to add.

**User:** News

**System:** Would you like to hear News from (content providers).

**User:** CNN

**System:** Would you like to hear (content details) or all.

**User:** National News

**System:** Your profile has been updated. Would you like to add more preferences.

**User:** No

**System:** Sprint Express, National News from CNN.

If the system fails to recognize the user at any point during the profile entry it should provide an error message and reprompt.

## **19.2. Grammars**

Users must have the ability to navigate between elements using the same grammar set as specified for the standard applications.

## **20. Bookmarks**

Users must have the ability to save navigation points using a Bookmark feature. If the user chooses to bookmark a site they must be prompted with a default title for the bookmark. Once the user has created a bookmark they must be able to hear a list of saved sites using the "Bookmarks" command. The user must also have the ability to navigate directly to a saved page by speaking the name of the bookmark from the Bookmarks sub-menu. The number of bookmarks must not exceed 50.

Users must have the ability to view and modify bookmarks on the web site. The name of the bookmark must be distinguishable from the primary grammar. A clash error should be presented to the user if the system is unable to differentiate the name of the bookmark from other menu items. Users must have the ability to modify the name and the URL of the bookmark.

## **21. Private Label Services**

The VB must have the ability to distinguish users based on the referring carrier. If the subscriber is not a SPCS customer the prompts and applications must be generic.

## **22. Premium Services**

SPCS must have the ability to create Premium Services. This will necessitate the ability to generate CDRs or billable log files based on the content provider. Premium Services must have the ability to be billed on a per use basis or through a MRC. The navigational options must include the ability to restrict access to major categories by user, dynamically modify the content providers for each category based on the subscription profile or calling plan of the user, and to notify the user that they will incur additional charges for accessing specific content. All premium services that incur per use charges will need to fall within designated price points. CDR development will not occur for each additional content provider.

## **23. Advertisements**

The VB must have the ability to insert advertisements as prompts prior to each content category and application. No advertisements will be played prior to the Ready prompt. The capability to prompt with advertisements must include random placement and placement based on criteria that include category matching and weighting. The advertisements may include links for more information on the product or service.

## **24. User Authentication**

The system must have the ability to authenticate the user. Several different options will need to be explored during development. Once the user has authenticated all applications must have the ability to verify the identity of the user based on the original authentication. The user must not be prompted to authenticate until he/she attempts to access an application that requires authentication. Users must have the ability to have several different authentication profiles

associated with a single ANI. Users should be given a maximum of three authentication attempts before being prompted with detailed help.

One option for authentication includes the creation of a Personal Identification Number (PIN). If the authentication includes a PIN then the PIN used to access content must be the same PIN used to access VC through the General Access Number (GAN). Users must have the ability to create the PIN on the web site. Users must not have the ability to personalize secure information, such as credit card numbers, without first establishing a PIN. If the user attempts to access a secure site through the voice interface without first creating a profile they must be prompted to create a PIN using the web site. The initial prompt for entering the PIN must include the ability to use DTMF for data entry.

The user must have the ability to log-out. If the user logs out they must hear a prompt confirming the log-out and then return to the Ready prompt.

## **25. URL Entry**

Users must have the ability to enter URLs on the voice and web interfaces. From the web interface the user will use the Bookmark application to enter URLs. On the voice interface the system must infer the "http://" prefix but not present an error if the user chooses to speak the protocol. All valid alphanumeric and standard keyboard characters must be recognized. No whole words will be recognized. The system should confirm the URL following 2 seconds of silence.

## **26. Text-to-Speech**

It is anticipated that multiple TTS engines will be deployed on this platform. The VB must include the capability to specify the TTS engine on a document and application level. This specification will most likely fall outside the standards of Voice XML. However, the method for specifying the different TTS engines must not invalidate the content providers Voice XML on other browsers. TTS engines may be segmented by application, male vs. female voice talents, and foreign languages. The specification of the available TTS engines to the content providers must include a clear indication of the strengths of each option and the recommended implementations. The parameter used to specify the TTS engine in the code must clearly distinguish the core functionality of that TTS (i.e., sprint\_spanish\_tts).

## **27. Coding Standards**

The first iteration of the VB must be compliant with Voice XML 1.0 specifications. All applications must be 1.0 compliant. Updates to the 1.0 specification must be supported within 4 months of final release by the W3C. All subsequent VB updates must be backward compatible and provide support for 1.0 and above.

## **28. Error Handling**

In general the system should provide an error message and reprompt if there is a mis-recognition error. Mis-recognition and Silence errors must include a minimum of three separate prompt states. Following two unsuccessful attempts the user should be given the option to get more information. All subsequent mis-recognition errors should include the menu options for that category with a DTMF equivalent.

Error handling must be performed at the document and application levels. However, the VC Root Application that is resident in the VB must include default error handling for the following conditions:

1. Error.badfetch – i.e., missing documents, bad URL, timeouts, malformed documents
2. Error.semantic – runtime errors

3. Error.noauthorization – user authentication error
4. Error.unsupported.format – i.e., grammar format, audio file, object type, MIME type
5. Error.unsupported.element – i.e., tag or element

## 29. Design and Testing

SPCS will contribute resources to continue the iteration of the design during the development of the VB. SPCS must have the ability to participate in the initial testing of the platform, prior to delivery in the STIC, to insure appropriate interpretation of these requirements.

## 30. Summary

It is the intention of SPCS to be the industry leader for platform integration and application design. These requirements are intended to represent the current standard of voice recognition technology and position SPCS above all other carriers with a 3Q delivery.

## 31. Glossary

**Barge-in** – The ability to speak and be recognized while the application is playing prompts or TTS.

**Disambiguation** – The ability to distinguish between two or more query results within a specified confidence threshold.

**Grammar** – A set of rules describing the technical structure of language. Identifies the words and phrases that will be understood by the system.

**Natural language** – Recognition permits un-scoped and continuous stream of input from the user

**Personas** – Style of communication established through prompts and dialog design. Does not necessarily imply personality.

---

**CLAIMS:**

What is claimed is:

1. A voice command platform comprising:

5 a user communication interface for communicating with users via a telecommunications

network;

a processor;

an application-processing module executable by the processor to process voice command applications, the voice command applications having navigation points, and the voice command 10 applications defining user-prompts, allowed grammars and application-logic, wherein the processor processes voice command applications during voice command sessions with users;

a user profile store including a navigation history record respectively for each of a plurality of users, the navigation history record for a given user identifying navigation points of voice command applications that the processor has processed during at least one voice command 15 session with the given user.

2. The voice command platform of claim 1, further comprising:

navigation-recording logic executable by the processor to record in the navigation history record for the given user an indication of a navigation point of a voice command application that 20 the processor has processed during a voice command session with the user.

3. The voice command platform of claim 1, wherein the navigation-recording logic is executable by the processor to record in the navigation history record for the given user each navigation point accessed during the voice command session with the user.

5 4. The voice command platform of claim 1, further comprising:

session-restore logic executable by the processor to restore a given voice command session with the given user based on the navigation history record for the given user.

10 5 The voice command platform of claim 4, wherein the session-restore logic is executable by the processor to determine that a system disconnect occurred during the given voice command session, and to thereafter restore the given voice command session.

15 6. The voice command platform of claim 5, wherein the session-restore logic is further executable by the processor to prompt the user for consent to restore the given voice command session.

7. The voice command platform of claim 5, wherein:

the user profile store includes an indication for the given user indicating that the system disconnect occurred; and

20 the session-restore logic is executable by the processor to determine, based on the indication, that the system disconnect occurred.

8. The voice command platform of claim 5, wherein the session-restore logic is executable to restore the given voice command session by a process comprising:  
using the navigation history record for the given user to identify a voice command application that the processor was processing at the time the system disconnect occurred; and  
5 loading and executing the voice command application.

9. The voice command platform of claim 5, wherein the navigation history lists navigation points in order of navigation.

10. 10. The voice command platform of claim 4, wherein the session-restore logic is executable to restore the given voice command session for a period of approximately 15 minutes after a system disconnect of the given voice command session.

15 11. The voice command platform of claim 4, wherein each of a plurality of the voice command applications are VXML applications, and each of a plurality of the navigation points are Universal Resource Indicators.

20 12. The voice command platform of claim 1, further comprising:  
expert-mode-transition logic executable by the processor to automatically transition the given user to expert-mode user status based on the navigation history record for the given user.

13. The voice command platform of claim 12, wherein the expert-mode-transition logic is executable:

to make a determination, based on the navigation history record for the given user, that the given user has accessed a navigation point at least a threshold number of times; and to set an expert-mode user flag in a profile record for the user, in response to the determination.

5

14. The voice command platform of claim 1, wherein the telecommunications network comprises a wireless communications link.

10 15. In a voice command platform of the type that communicates with users via a telecommunications network and that executes voice command applications during voice command sessions with users, a method comprising:

15 storing, respectively for each of a plurality of users, a navigation history log indicating navigation points of voice command applications that the platform has executed during at least one voice command session with the user.

15

16. The method of claim 15, further comprising:

using the navigation history log to restore a previous voice command session with the user.

20

17. The method of claim 16, wherein using the navigation history log to restore a previous voice command session with the user comprises:

determining that a system disconnect occurred from the previous voice command session;

identifying, based on the navigation history log, a given navigation point of a given voice command application that the platform was executing at the time the system disconnect occurred;

loading the given voice command application from the given navigation point; and  
5 executing the given voice command application.

18. The method of claim 17, further comprising restoring the previous voice command session with the user at the initiation of a subsequent voice command session with the user.

10

19. The method of claim 18, further comprising prompting the user for consent to restore the previous voice command session.

20. The method of claim 15, further comprising:

15 using the navigation history log to determine that the user should be automatically transitioned to expert-mode user status; and  
automatically transitioning the user to expert-mode user status.

21. The method of claim 20, wherein:

20 using the navigation history log to determine that the user should be automatically transitioned to expert-mode user status comprises using the navigation history log to determine that the user should be automatically transitioned to expert-mode user status with respect to a given navigation point; and

---

automatically transitioning the user to expert-mode user status comprises automatically transitioning the user to expert-mode user status with respect to the given navigation point.

22. The method of claim 20, wherein using the navigation history log to determine  
5 that the user should be automatically transitioned to expert-mode user status comprises:

determining, based on the navigation history log, that a given navigation point has been accessed at least a threshold number of times during at least one voice command session with the user; and

10 responsively determining that the user should be automatically transitioned to expert-mode user status with respect to at least the given navigation point.

23. The method of claim 15, wherein the telecommunications network comprises a wireless communication link.

15 24. A voice command platform comprising:

a processor;

stored indications, respective for each of a plurality of users, of a use-level of the user and a navigation history of the user; and

20 logic executable by the processor to switch the use-level of a user from one level to another based on the navigation history of the user.

25. The voice command platform of claim 24, wherein:

the use-level is selected from the group consisting of (i) expert-mode and (ii) not-expert-mode; and

the logic is executable by the processor to automatically switch the use-level of the user from non-expert mode to expert-mode, based on the navigation history of the user.

5

26. The voice command platform of claim 23, wherein the logic is further executable by the processor to prompt the user for authority to switch the user's use-level.

10

27. The voice command platform of claim 23, further comprising:

a voice command application including expert-mode logic and non-expert-mode logic, wherein the processor executes the expert-mode logic when the voice command platform is interacting with a user for whom the user profile store specifies an expert-mode use-level, and the processor executes the not-expert-mode logic when the voice command platform is interacting with a user for whom the user profile store specifies a not-expert-mode use-level.

15

28. In a voice browser system arranged to execute voice-tag applications and to interface between voice tag applications and users, a method comprising:

maintaining a navigation-history record that indicates a user's navigation history through at least one of the voice-tag applications via the voice browser system;

20

maintaining a use-mode record that indicates whether the user is an expert-user of the at least one voice-tag application;

automatically setting the use-mode record to indicate that the user is an expert-user of the at least one voice-tag application, based on the navigation-history record; and

when executing the at least one voice-tag application, interfacing with the user according to the use-mode record.

29. The method of claim 28, wherein the at least one voice-tag application defines a  
5 standard set of logic including a standard set of voice prompts and the at least one voice-tag application defines an expert set of logic including an expert set of voice prompts, and wherein interfacing with the given user according to the use-mode record comprises:

making a determination that the use-mode record indicates that the user is an expert-user of the at least one voice-tag application; and  
10 responsive to the determination, executing the expert set of logic rather than the standard set of logic.

30. The method of claim 29, wherein voice prompts of the expert set are shorter in duration than voice prompts of the standard set.

15

31. The method of claim 29, wherein the standard set of voice prompts includes a voice prompt for a given menu item, and the expert set of voice prompts includes a tone prompt for the given menu item.

20

32. The method of claim 31, wherein automatically setting the use-mode record to indicate that the user is an expert-user of the at least one voice-tag application, in response to the navigation-history record, comprises:

determining that the user has accessed the given menu item at least a threshold number of times, and responsively setting the use-mode record to indicate that the user is an expert-user of the at least one voice-tag application.

5        33.      The method of claim 28, wherein the at least one voice-tag application defines a standard prompt for a given menu item and an expert prompt for the given menu item, and wherein interfacing with the given user according to the use-mode record comprises:

making a determination that the use-mode record indicates that the user is an expert-user of the at least one voice-tag application; and

10        responsive to the determination, executing the expert prompt rather than the standard prompt.

15        34.      The method of claim 33, wherein the expert prompt is shorter in duration than the standard prompt.

35.      The method of claim 34, wherein the standard prompt is a voice prompt and the expert prompt is a tone prompt.

-----Original Message-----

From: Aaronson [<mailto:Aaronson@mbhb.com>]  
Sent: Tuesday, July 31, 2001 5:02 PM  
To: Packingham, Kevin D. PCS.  
Cc: Aaronson; steven.j.funk; Timothy.a.nehls  
Subject: Patent Matter -- Sprint Docket Nos. 1700, 1730, 1731 and 1732

CONFIDENTIAL & PRIVILEGED  
ATTORNEY-CLIENT PRIVILEGE

Kevin,

I have now prepared a draft consolidated written description (with draft drawings) that will form the basis for the four patent applications that we plan to file regarding your voice command improvements. Attached is the draft description for your review.

Note that each patent application will have a common description but will have a unique cover sheet, a unique set of claims and probably a unique summary section and abstract. I have not yet completed the claims or written the summaries or abstracts. So the attached document excludes those components. However, the attached document includes cover sheets for all four applications.

Please review the draft description, and provide me with your comments at your earliest convenience. Meanwhile, I will be continuing with work on the claims and other aspects of the applications. When I complete the claims, I will forward them to you for your review as well.

Please let me know if you have any questions.

Thanks.

Larry H. Aaronson  
McDonnell Boehnen Hulbert & Berghoff  
300 South Wacker Drive  
Chicago, Illinois 60606

312-913-0001 phone  
312-913-2141 direct  
312-913-0002 fax

<<1700 et seq. draft description.doc>>

**SPRINT CORPORATION**  
**VOICE COMMAND PLATFORM**  
**IMPROVEMENTS**  
**DRAFT PATENT APPLICATIONS**  
*(excluding claims)*

**JULY 30, 2001**

**SPRINT DOCKET NOS.**  
**1700, 1730, 1731, 1732**

*Confidential & Privileged  
Attorney-Client Privilege*

**SPECIFICATION**  
**(Sprint Docket No. 1700)**

**TO ALL WHOM IT MAY CONCERN:**

Be it known that we, **Kevin PACKINGHAM**, a citizen of \_\_\_\_\_ and resident of \_\_\_\_\_, **Elizabeth ROCHE**, a citizen of \_\_\_\_\_ and resident of Prairie Village, Kansas, and **Terry T. YU**, a citizen of the United States and resident of Overland Park, Kansas, have invented a new and useful:

**METHOD AND SYSTEM FOR  
EHNANCED USER CONTROL  
OVER A VOICE COMMAND PLATFORM**

the following of which is a specification.

**RELATED APPLICATIONS**

This application is related to the following commonly owned applications filed on the same date: (i) "Method and System for Enhanced Application Control Over a Voice Command Platform," naming Kevin Packingham as inventor, (ii) "Method and System for Monitoring and Use of Session Information in a Voice Command Platform," naming Kevin Packingham and Elizabeth Roche as co-inventors, and (iii) "Method and System for Unified Message Notification in a Voice Command Platform," naming Kevin Packingham and Robert W. Hammond as co-inventors. The entirety of each of these other applications is hereby incorporated by reference.

**SPECIFICATION**  
**(Sprint Docket No. 1730)**

**TO ALL WHOM IT MAY CONCERN:**

Be it known that I, **Kevin PACKINGHAM**, a citizen of \_\_\_\_\_ and resident of Overland Park, Kansas, have invented a new and useful:

**METHOD AND SYSTEM FOR  
ENHANCED APPLICATION CONTROL  
OVER A VOICE COMMAND PLATFORM**

the following of which is a specification.

**RELATED APPLICATIONS**

This application is related to the following commonly owned applications filed on the same date: (i) "Method and System for Enhanced User Control Over a Voice Command Platform," naming Kevin Packingham, Elizabeth Roche and Terry T. Yu as co-inventors, (ii) "Method and System for Monitoring and Use of Session Information in a Voice Command Platform," naming Kevin Packingham and Elizabeth Roche as co-inventors, and (iii) "Method and System for Unified Message Notification in a Voice Command Platform," naming Kevin Packingham and Robert W. Hammond as co-inventors. The entirety of each of these other applications is hereby incorporated by reference.

**SPECIFICATION**  
**(Sprint Docket No. 1731)**

**TO ALL WHOM IT MAY CONCERN:**

Be it known that we, **Kevin PACKINGHAM**, a citizen of \_\_\_\_\_ and resident of Overland Park, Kansas, and **Elizabeth ROCHE**, a citizen of \_\_\_\_\_ and resident of Prairie Village, Kansas, have invented a new and useful:

**METHOD AND SYSTEM FOR  
MONITORING AND USE OF SESSION INFORMATION  
IN A VOICE COMMAND PLATFORM**

the following of which is a specification.

**RELATED APPLICATIONS**

This application is related to the following commonly owned applications filed on the same date: (i) "Method and System for Enhanced User Control Over a Voice Command Platform," naming Kevin Packingham, Elizabeth Roche and Terry T. Yu as co-inventors, (ii) "Method and System for Enhanced Application Control Over a Voice Command Platform," naming Kevin Packingham as inventors, and (iii) "Method and System for Unified Message Notification in a Voice Command Platform," naming Kevin Packingham and Robert W. Hammond as co-inventors. The entirety of each of these other applications is hereby incorporated by reference.

**SPECIFICATION**  
**(Sprint Docket No. 1732)**

**TO ALL WHOM IT MAY CONCERN:**

Be it known that we, **Kevin PACKINGHAM**, a citizen of \_\_\_\_\_ and resident of Overland Park, Kansas, and **Robert W. HAMMOND**, a citizen of the United States and resident of Overland Park, Kansas, have invented a new and useful:

**METHOD AND SYSTEM FOR  
UNIFIED MESSAGE NOTIFICATION  
IN A VOICE COMMAND PLATFORM**

the following of which is a specification.

**RELATED APPLICATIONS**

This application is related to the following commonly owned applications filed on the same date: (i) "Method and System for Enhanced User Control Over a Voice Command Platform," naming Kevin Packingham, Elizabeth Roche and Terry T. Yu as co-inventors, (ii) "Method and System for Enhanced Application Control Over a Voice Command Platform," naming Kevin Packingham as inventors, and (iii) "Method and System for Monitoring and Use of Session Information in a Voice Command Platform," naming Kevin Packingham and Elizabeth Roche as co-inventors. The entirety of each of these other applications is hereby incorporated by reference.

## **BACKGROUND**

### **1. Field of the Invention**

The present invention relates to telecommunications and, more particularly, to functions of a voice command platform.

### **2. Description of Related Art**

A voice command platform provides an interface between speech communication with a user and computer-executed voice command applications. Generally speaking, a person can call a voice command platform from any telephone and, by speaking commands, can browse through navigation points (e.g., applications and/or menus items within the applications) to access and communicate information. The voice command platform can thus receive spoken commands from the user and use the commands to guide its execution of voice command applications, and the voice command platform can "speak" to a user as dictated by logic in voice command applications.

For instance, a person may call a voice command platform, and the platform may apply a voice command application that causes the platform to speak to the user, "Hello. Would you like to hear a weather forecast, sports scores, or stock quotes?" In response, the person may state to the platform, "weather forecast." Given this response, the application may cause the platform to load and execute a subsidiary weather forecasting application. The weather forecasting application may direct the platform to speak another speech prompt to the person, such as "Would you like to hear today's weather or an extended forecast?" The person may then respond, and the weather forecasting application may direct the voice command platform to execute additional logic or to load and execute another application based on the person's response.

A robust voice command platform should therefore be able to (i) receive and recognize speech spoken by a user and (ii) provide speech to a user. The platform can achieve these functions in various ways.

On the incoming side, for instance, the platform may include an analog-to-digital (A-D) converter for converting an analog speech signal from a user into a digitized incoming speech signal. (Alternatively, the user's speech signal might already be digitized, as in a voice-over-IP communication system, for instance, in which case A-D conversion would be unnecessary). The platform may then include a speech recognition (SR) engine, which functions to analyze the digitized incoming speech signal and to identify words in the speech. The SR engine will typically be a software module executable by a processor.

Usually, a voice command application will specify which words or "grammars" a user can speak in response to a prompt for instance. Therefore, the SR engine will seek to identify one of the possible spoken responses. (Alternatively, the SR engine may operate to identify any words without limitation).

In order to identify words in the incoming speech, the SR engine will typically include or have access to a dictionary database of "phonemes" (small units of speech that distinguish one utterance from another). The SR engine will then analyze the waveform represented by the incoming digitized speech signal and, based on the dictionary database, will determine whether the waveform represents particular words. For instance, if a voice command application allows for a user to respond to a prompt with the grammars "sales," "service" or "operator", the SR engine may identify the sequence of one or more phonemes that makes up each of these grammars respectively. The SR engine may then analyze the waveform of the incoming digitized speech signal in search of a waveform that represents one of those sequences of

phonemes. Once the SR engine finds a match, the voice command platform may continue processing the application in view of the user's spoken response.

Additionally, the SR engine or an ancillary module in the voice command platform will typically function to detect DTMF tones dialed by a user and to convert those DTMF tones into representative data for use in the execution of a voice command application. Thus, for instance, a voice command application might define a particular DTMF grammar as an acceptable response by a user. Upon detection of that DTMF grammar, the platform may then apply associated logic in the application.

On the outgoing side, the voice command platform may include a text-to-speech (TTS) engine for converting text into outgoing digitized speech signals. In turn, the platform may include a digital-to-analog (D-A) converter for converting the outgoing digitized speech signals into audible voice that can be communicated to a user. (Alternatively, the platform might output the digitized speech signal itself, such as in a voice-over-IP communication system).

A voice command application may thus specify text that represents voice prompts to be spoken to a user. When the voice command platform encounters an instruction to speak such text, the platform may provide the text to the TTS engine. The TTS engine may then convert the text to an outgoing digitized speech signal, and the platform may convert the signal to analog speech and send it to the user. In converting from text to speech, the TTS engine may also make use of the dictionary database of phonemes, so that it can piece together the words that make up the designated speech.

Also on the outgoing side, a voice command platform may include a set of stored voice prompts, in the form of digitized audio files (e.g., \*.wav files) for instance. These stored voice prompts would often be common prompts, such as "Hello", "Ready", "Please select from the

following options", or the like. Each stored voice prompt might have an associated label (e.g., a filename under which the prompt is stored). And, by reference to the label, a voice command application might specify that the voice command platform should play the prompt to a user. In response, the voice command platform may retrieve the audio file, convert it to an analog waveform, and send the analog waveform to the user.

A voice command application can reside permanently on the voice command platform, or it can be loaded dynamically into the platform. For instance, the platform can include or be coupled with a network or storage medium that maintains various voice command applications. When a user calls the platform, the platform can thus load an application from the storage medium and execute the application. Further, in response to logic in the application (such as logic keyed to a user's response to a menu of options), the platform can load and execute another application. In this way, a user can navigate through a series of applications and menus in the various applications, during a given session with the platform.

A voice command application can be written in any of a variety of computer languages. One such language is VoiceXML (or simply "VXML"), which is a tag-based language similar to the HTML language that underlies most Internet web pages. (Other analogous languages, such as SpeechML and VoxML for instance, are available as well.) By writing a voice command application in VXML, the application can thus be made to readily access and provide web content, just as an HTML-based application can do. Further, when executed by the voice command platform, the VXML application can effectively communicate with a user through speech.

In order for a voice command platform to execute a VXML application or other tag-based application, the platform should include a VXML browser or "interpreter." The VXML

interpreter functions to interpret tags set forth in the application and to cause a processor to execute associated logic set forth in the application.

A VXML application can be made up of a number of VXML documents, just like an HTML web site can be made up of a number of HTML pages. A VXML application that is made up of more than one document should include a root document, somewhat analogous to an HTML home page. According to VXML, the root document defines variables that are available to all subsidiary documents in the application. Whenever a user interacts with documents of a VXML application, the root document of the application is also loaded. Therefore, variables defined in the root document should be available during execution of any of the documents of the application.

Each VXML document will include a <vxml> tag to indicate that it is a VXML document. It may then include a number of <form> sections that can be interactive (e.g., prompting a user for input) or informational (e.g., simply conveying information to a user.) Within a given form, it may further include other executable logic.

A VXML document can also define grammars as described above. In particular, VXML grammars are words or terms that the VXML application will accept as input during execution of the application. When a VXML application is executed on a voice command platform, the platform may provide the SR engine with an indication of the grammars that the VXML application will accept. Once the SR engine detects that a user has spoken one of the grammars, the platform may apply that grammar as input to the VXML application, typically proceeding to execute a set of logic (e.g., a link to another document) in response.

For example, a VXML document can define as grammars a number of possible options, as well as a number of possible words that a user can speak to select those options. For instance,

a document might define as options of clothing the items "hat", "shirt", "pants" and "shoes". In turn, the document might define the following as acceptable grammars for the "hat" option: "hat", "visor", "chapeaux" and "beret".

Grammars defined in the root document of a VXML application are, by default, available for use in all of the subsidiary documents of the application. Thus, when a voice command platform is executing a VXML application, if a user speaks a grammar that is defined in the root document of the application, the voice command platform should responsively execute the logic that accompanies that grammar in the root document of the application.

In a voice command platform, each navigation point may have a respective identifier or label. For example, each voice command application can have a respective label, such as a network address where the application is maintained. As another example, a voice command application can define a number of successive menus through which a user can browse, and each menu might have a respective label by which it can be referenced. A voice command platform can use these labels to move from application to application or from menu item to menu item, just as hyperlinks operate to cause a browser to move from one web page (or component of one web page) to another.

In VXML, for instance, each VXML document will have a respective Universal Resource Identifier (URI), which is akin to a Universal Resource Locator (URL) used to identify the network location of an HTML page. A given VXML document may thus define logic that instructs the voice command platform to load and execute another VXML document from a designated URI. For instance, a VXML document may indicate that, if a user speaks a particular grammar, the platform should load and execute a particular VXML document from a designated

URI, but that, if the user speaks another grammar, the platform should load and execute another VXML document from another designated URI.

An example of a VXML application is a weather reporting application. The weather reporting application may have a root document that includes a tag defining a welcome message and prompting a user to indicate a city or zip code. The root document may further set forth a bundle of grammars that are possible city names and corresponding zip codes that a user can speak in response to the prompt.

When the voice command platform executes this root document, it may thus send the welcome message/prompt to the TTS engine to have the message/prompt spoken to the user. In turn, when the user speaks a response, the SR engine would identify the response as one of the acceptable grammars. The platform would then continue executing the root document in view of the spoken response.

The root document might next prompt the user to indicate whether the user would like to hear today's weather or an extended forecast, and the user would again speak a response. In turn, the root document might indicate that, if the user selects "today's weather," the platform should load and execute a subsidiary document from a designated URI, and if the user selects "extended forecast," the platform should load and execute a different subsidiary document from another designated URI. Of course, many other examples of VXML applications are possible as well.

In most cases, a platform provider will own and operate the voice command platform. Content providers will then provide the VXML applications to be executed by the platform. The platform provider may also provide some applications for the platform and may therefore function as a content provider as well.

Further, a content provider can personalize a VXML application, through reference to user profiles. For example, a telecommunications service provider (e.g., local exchange carrier or interexchange carrier) can provide a voice-activated-dialing (VAD) application that allows users to dial a telephone number by speaking a name. To support this feature, the VAD application may direct the voice command platform to prompt a user for a user ID or to determine the user ID based on calling number identification provided when the user's call was connected to the platform. The VAD application may then instruct the platform to call up a personalized VAD application (through use of Microsoft Active Server Pages, for instance), which is tied to the user's personal address book. Each name in the address book may then define an acceptable grammar. When the user speaks one of the names, the application may cause the platform to retrieve a corresponding telephone number and to provide that number to a network switch to facilitate initiating the call.

## SUMMARY

## BRIEF DESCRIPTION OF THE DRAWINGS

Exemplary embodiments of the present invention are described herein with reference to the drawings, in which:

Figure 1 is a functional block diagram illustrating the layers of a system in which the exemplary embodiments can be employed; and

Figure 2 is a functional block diagram of a voice command platform including enhanced system logic in accordance with the exemplary embodiments.

## **DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS**

### **1. Exemplary Voice Command System**

Referring to the drawings, Figures 1 is a functional block diagram illustrating the layers of a system in which an exemplary embodiment of the present invention can be employed. As shown in Figure 1, the system may be considered to include three layers, (i) a user layer 12, (ii) a platform or system layer 14, and (iii) an application layer 16. The user layer 12 provides a mechanism 18, such as a telephone, for a person to communicate by voice with the platform. The system layer, in turn, includes a user communication interface 20, a processor 22 (i.e., one or more processors), a voice translator module 24, a VXML interpreter module 26, and cache 28. Application layer 16 then defines an application 30, which may be made up of a root document 32 and subsidiary documents 34 that can be loaded into cache 28 and executed by processor 22, i.e., by the voice command platform.

User communication interface 20 may take various forms. For example, the user communication interface can provide a circuit or packet interface with a telecommunications network such as the PTSN or the Internet. The communication interface may, in turn, include an A-D and D-A converter (not shown) as described above, for converting between analog signals on the user side and digital signals on the platform side. Processor 22 then sends and receives communications via user communication interface 20.

Voice translator module 24 and VXML interpreter module 26 preferably define program instructions that can be executed by processor 22 and data that can be referenced by the processor, to carry out basic voice platform functions. All of this program logic can be stored in suitable data storage, such as ROM or a disk drive for example.

For instance, voice translator module 24 preferably defines an SR engine 36 and a TTS engine 38, as well as a voice prompt store 40. Further, voice translator module 24 may include a phoneme dictionary 42 that the processor can reference to facilitate operation of the SR engine and the TTS engine. VXML interpreter module 26, in turn, may facilitate execution of application 30, such as by interpreting tags in the application.

Cache 28 functions to temporarily hold application 30 (or one or more documents of the application) when processor 22 is executing the application. Thus, in operation, processor 22 may retrieve application 30 from a designated URI on the Internet (or elsewhere) and may load the application into cache 28. The processor may then execute the application, using VXML interpreter 26 to interpret tags, using TTS engine 38 and voice prompt store 40 to send speech to a user, and using SR engine 36 to recognize speech spoken by a user.

It should be understood that that this and other arrangements described herein are set forth for purposes of example only. As such, those skilled in the art will appreciate that other arrangements and other elements (e.g., machines, interfaces, functions, orders and groupings of functions, etc.) can be used instead, and some elements may be omitted altogether. Further, as in most telecommunications applications, those skilled in the art will appreciate that many of the elements described herein are functional entities that may be implemented as discrete or distributed components or in conjunction with other components, and in any suitable combination and location.

For example, although the voice command system is shown to include a VXML interpreter, the system may include another type of voice browser. Alternatively, the system might not be browser-based at all. In that event, some of the functions described herein could be applied by analogy in another manner, such as through more conventional interactive voice response (IVR) processing. Other examples are possible as well.

## 2. Enhanced System Layer

In accordance with the exemplary embodiment, system layer 14 can be enhanced in several respects, so as to provide a more robust voice command platform. Figure 2 depicts an expanded functional block diagram of the system layer 14, showing some of the improvements contemplated.

As shown by way of example in Figure 2, exemplary system layer 14 includes the components described above, such as a user communication interface 20, a processor 22, voice translator 24, VXML interpreter 26, and cache 28. In addition, according to the exemplary embodiment, the system layer includes or has access to an enhanced set of system logic, preferably comprised of software modules that can be executed by the processor and data that can be referenced by the processor.

Some or all of the enhanced system logic can be co-located with the platform or can be located elsewhere, such as on an accessible network. Further, aspects of this enhanced system logic may be integrated together with the voice translator module 24 and/or with the VXML interpreter module 26 to any extent desired.

Referring in more detail to Figure 2, the enhanced system logic can include multiple text-to-speech engines ( $TTS_1, TTS_2, \dots, TTS_N$ ) 44 (e.g., separate sub-routines of a main TTS module, or separate TTS modules altogether) that may be selectively applied by processor 22 (i.e.,

selected by the processor in response to a designated stimulus or instruction). The TTS engines can differ from each other in terms of language (e.g., English, Spanish, etc.), dialect (e.g., accent), voice (e.g., male, female) and/or other aspects, so as to speak to users in different manners. (In the exemplary embodiment, the platform can similarly include multiple speech recognition engines (not shown), or multiple core phoneme dictionaries, to facilitate converting speech of various types to data.)

Additionally, the enhanced system logic can include multiple voice prompt stores or sets of voice prompts 46 that can be selectively applied by the processor. Each prompt set may differ from the other prompt sets in various ways, such as being spoken in different dialects or different voices in a common language, or being spoken in different languages, or with different background music, for instance.

For example, one prompt set might comprise the prompts spoken in a particular celebrity voice (e.g., in a recognizable voice of a particular movie or television star), and another prompt set might comprise the prompts spoken in a different celebrity voice. Still another prompt set might comprise the prompts spoken in a male voice, and another might comprise the prompts spoken in a female voice. Further, another might comprise the prompts spoken in a high energy voice, while another might comprise the prompts spoken in a low key voice. And still another might comprise the prompts spoken in English, while another might comprise the prompts spoken in French.

On the platform, the prompt sets can be stored in separate directories (folders) on a disk drive, where each directory has a respective name to designate the prompt set. For instance, the directories may be named "Celebrity1", "Celebrity2", "male", "female", "high energy", "low

key," etc. This can facilitate quick and easy retrieval of desired prompts. Other mechanisms for storing and identifying the various prompt sets, however, are possible as well.

As further shown in Figure 2, the enhanced system logic can include a set of one or more secondary phoneme dictionaries 48. Each of these secondary dictionaries is preferably an application-specific (or content-provider-specific) phoneme dictionary, provided by a content provider to improve voice translation functions when the platform is executing the respective application.

Additionally, the enhanced system logic can include a set of system-level grammars 50. The system-level grammars 50 are preferably grammars that are defined for the voice command platform generally. As such, the voice command platform can recognize the system-level grammars when spoken by a user at substantially any navigation point, i.e., while any voice command application is being executed by the platform.

In addition, the enhanced system logic can include a user profile store 52. For various aspects of the exemplary embodiment, the user profile store 52 may define (i) user ID information 54, (ii) user preference information 56, (iii) user session information 58, and (iv) user consolidated-messaging information 60. This information can be maintained in a relational database or in any other manner desired (whether as one module or separate objects, such as a meta-directory for instance), and it can be co-located with the rest of the platform or can be located elsewhere.

User ID information 54 may include any information that the platform can use to identify a user, so as to facilitate providing personalized voice command services across multiple applications. In the exemplary embodiment, user ID information will be ANI information, such

as the billing number or telephone number of the calling party. Alternatively, however, user ID information can also take other forms, such as username and password.

User preference information 56 can also take various forms. For example, user preference information may include a set of personal grammars/aliases, which define grammars specific to a user. As another example, user preference information may include an expert mode flag (or flags) for the user, which indicates whether or not the platform should interface with the user according to an "expert mode." As still another example, user preference information may include indications of preferred content providers, such as a preferred e-mail provider, a preferred voice-mail provider, a preferred weather-reporting provider, etc.

As yet another example, user preference information may include a set of bookmarks/favorites (with associated names), which function as user-specific shortcuts to URIs or menu items. And as still a further example, user preference information may include a user-specific speech persona, which indicates a voice or persona for the voice translator to use in communicating to the user (e.g., an indication of which of several possible pre-recorded versions of a prompt to play to the user and/or which TTS engine to use for the user).

User session information 58 can also take various forms. For example, user session information can comprise a record of a user's current session state. In this regard, session state information may include an indication of whether or not the voice translator (or, more specifically, the speech recognition engine) is currently muted for the user's session. And as still another example, session state information may include an indication of which of the multiple TTS engines 44 and/or voice prompt sets 46 is currently active (selected, or being used) for the user or for the user's session.

User session information 58 can also comprise a record of a user's navigation history. For example, the navigation history record may specify in order the last predetermined number of navigation points (e.g., URIs, menu items, etc.) that the user has accessed, preferably including the latest navigation point accessed by the user. Further, the navigation history record may include a summary ("rollup") indication of the number of times that the user has accessed each navigation point. This navigation history information can be used as a basis to determine whether the system should automatically switch the user to expert mode, as well as to restore a user's voice command session in the event a system disconnect occurs.

User consolidated-messaging information 60, similarly, can take various forms. In the exemplary embodiment, the consolidated-messaging information comprises a summary the number of messages waiting for the user at multiple separate message portals. For instance, the consolidated-messaging information for a user can indicate a count of voice mail messages waiting for the user at one message portal, a count of e-mail messages waiting for the user at another message portal, and account of fax messages waiting for the user at another message portal. The user profile may also identify each of the user's message accounts (e.g., POP3 e-mail account information, etc.)

As still further shown in Figure 2, enhanced system logic preferably defines a number of additional logic modules 62 that are executable by the processor to carry out enhanced functions described herein. These enhanced functions can involve, for instance, (a) enhancing user control over the voice command platform, (b) enhancing application control over the voice command platform, (c) monitoring and use of navigation history, and (d) providing unified messaging notification. Each of these enhanced functions will be described in more detail in the following sections.

a. **Enhancing User-Specific Control Over the Voice Command Platform**

In accordance with the exemplary embodiment, the enhanced system logic provides for enhancing user control over the voice command platform, preferably across multiple applications. Generally speaking, the enhanced control involves having the platform receive a user-specification and perform a function in response to the user-specification. The user-specification may be one or more parameters indicated, for instance, by the user profile store 52 (whether specified in advance by the user, an administrator, or otherwise) and/or by a command submitted by a user during a session with the voice command platform. The function performed in response can also take various forms but is generally a system-layer function. That is, the platform performs a function that is applicable across substantially all voice command applications that the platform executes.

As examples, enhanced user control over the voice command platform can involve (i) recognizing system-level grammars (i.e., system-specific grammars) that can be spoken by a user, (ii) recognizing personal grammars (i.e., user-specific aliases of other grammars) that can be spoken by a user, (iii) hosting a set of personal bookmarks (i.e., user-specific shortcuts to navigation points), (iv) muting the SR engine 36 in response to a command from the user, and (v) selectively applying a user-specific speech persona (i.e., a persona set by or for voice command sessions with the user).

Many of these user-control functions may involve having processor 22 refer to user profile store 52 in order to determine a user-specification. In order to facilitate this, a mechanism should preferably be provided to identify a user who contacts the platform. This mechanism can take various forms.

As noted above, for example, user profile store 52 preferably indicates as user ID information each user's ANI. When a user places a conventional telephone call to the voice command platform, the processor can receive the user's ANI in the call setup signaling or otherwise. The processor may then query the user profile store 52 to find the user's profile record. In turn, the processor can store in memory a pointer to the user's profile record. Alternatively, the processor can extract parameters from the user's profile record and store those parameters in memory for quicker access.

As another example, as also noted above, user profile store 52 could indicate a username and password for each user. In an alternative arrangement, the processor could be programmed to prompt a user to specify a username when the user calls the platform, and the processor can then query the user profile store to find a matching record. Further, the processor can prompt the user to specify a password, and the processor can then authenticate the user by reference to the stored password in the profile record.

This alternative identification function can be carried out by having the processor execute a system-level VXML application when a caller calls the platform. The system-level application may define a username voice prompt such as "Welcome to the voice command system. Please state your name to begin." In turn, the system-level application can receive a response spoken by the user and can direct the processor to search the user profile for a matching record. The system-level application

Each of the enhanced user-control functions will now be described.

i. **Responding to User Utterance  
of System-Level Grammars**

Normally, an *application* that is executed by the voice command platform can define a set of grammars that the system layer will recognize. For instance, a weather reporting application might define "today's weather" and "extended forecast" as grammars that a user can speak, and the application might then define logic to be executed respectively in response to each of those grammars.

In accordance with the exemplary embodiment, as noted above, enhanced system logic 40 can include a set of system layer grammars, which are grammars that the system layer (e.g., VXML interpreter) will recognize globally, i.e., substantially regardless of what application the system is currently executing.

The system-level grammars, and their associated logic, can take various forms. For example, the processor can be programmed to recognize as system-level grammars the terms "Voice Command" and "Main Menu" and to respond to either of these grammars by presenting the user with the initial ready prompt, i.e., the prompt that the user might first encounter when the user calls the platform. As another example, the processor can be programmed to recognize as a system-level grammar the word "Mute" and to respond to this grammar by turning off (or partially turning off) the speech recognition engine 36. Similarly, the processor can be programmed to recognize the word "Unmute" and to respond to this grammar by turning on the speech recognition engine 36.

As still another example, the processor can be programmed to recognize the grammars "Save" and "Bookmark" and to respond to either of these grammars by saving in the user's profile a bookmark to the current navigation point, or by executing a subsidiary bookmark-

application. (As will be noted below, the processor may prompt the user to specify a shortcut name for the bookmark, which the processor may also store.) And as yet another example, the processor may be programmed to recognize the grammar "Voice Command Help" and to respond to this grammar by executing a voice command help module/application. Still another example may be a "Pause" or "Go to Sleep" system-level grammar, which may direct the processor to pause the current session.

In the exemplary embodiment, the system-level grammars and their associated logic can be defined in a system-level application that gives the system-level grammars a global scope for subsidiary applications. For example, the SR engine can be hard-coded to recognize the set of system-level grammars. Alternatively, in a VXML-based platform for instance, the system-level application can be a root VXML application that the platform is programmed to run by default when a user calls the platform (preferably without having to be referenced by another VXML document).

The root VXML application can function to welcome the user to the platform, such as by providing an initial ready prompt to the user and providing an initial menu of choices (such as VAD, news, sports, stocks, etc.) and calls to subsidiary applications. And the root VXML application can ideally define the system-level grammars and their associated logic. Thus, as subsidiary applications are called from the root document and from subsidiary documents, the system-level grammars and associated logic defined in the root VXML application will remain available for use in all subsidiary documents.

Preferably, if a content provider defines a grammar in an application that conflicts with (e.g., is the same as) a system-level grammar, the system-level grammar will take precedence. I.e., platform would be programmed to execute the logic defined for the system-level grammar,

rather than the logic defined for the conflicting application-level grammar. However, exceptions could be possible.

As another example, the system layer can be programmed with one or more system-level grammars that function as "aliases," i.e., nicknames or shortcuts for other grammars. For instance, the system layer might include in memory a table that translates between alias system-level grammars and actual grammars. When a user speaks one of the alias system-level grammars, the system layer can find a match in the table and responsively convert the spoken grammar into the associated actual grammar. The system-layer can then treat the actual grammar as having been spoken by the user.

Advantageously, because this aliasing mechanism exists in the system layer, the voice command platform can provide translations between aliases and actual grammars globally for all applications (or, allowing for exceptions, substantially all applications) executed by the platform. For example, the system-layer might define the terms "weather outlook" and "tomorrow's weather" as alias grammars for the word "extended forecast." In turn, when the system is executing a weather-reporting application and a user speaks "weather outlook," the system may convert "weather outlook" into "extended forecast" and treat "extended forecast" as being spoken by the user. The system may thus execute the logic that the application defines for the response "extended forecast," such as calling up an extended-forecast subsidiary URI, for instance.

## **ii. Responding to User Utterance of Personal-Grammars**

In the exemplary embodiment, as noted above, user profile store 52 can also define a set of user-specific grammars, i.e., personal grammars. In the exemplary embodiment, these

personal grammars function as aliases for other grammars, similar to the system-level alias grammars described above.

Thus, a user's profile preferably includes a table or other mechanism that translates between personal grammars and actual grammars. When a user speaks a personal grammar, the system can then find a match in the table and responsively convert the spoken grammar into the associated actual grammar. The system layer can then treat the actual grammar as having been spoken by the user.

As an example, an application may define "stocks" as a grammar that a user can select to go to a stock-reporting application. Advantageously, the user's profile may define the personal grammar "check portfolio" to correspond to the grammar "stocks." When the user speaks "check portfolio," the system can thus translate the grammar into "stocks" and proceed to execute the application as if the user has spoken "stocks." Other examples are possible as well.

### **iii. Allowing User-Control over SR Engine**

In an exemplary voice command platform, processor 22 is programmed to be able to selectively turn on or off the SR engine 36. When the SR engine is turned off, the system will not recognize words spoken by the user or will recognize only very specific grammars (or a limited set of grammars). For instance, the SR engine may be set to simply not analyze incoming digitized speech waveforms, or the SR engine may be set to analyze only the digitized speech waveforms in search of only very specific grammars.

Advantageously, enhanced system logic 40 enables a user to control the state of the speech recognition engine during the user's session with the voice command platform. As described above, for instance, a system-level grammar such as "Mute" can be defined. When the user says "Mute," the processor may respond by setting a flag in memory to indicate that the SR

engine is turned off. The SR engine module 36 may then include logic to indicate that, if the flag indicates the SR is turned off, the processor should not apply the SR engine module or should apply only designated aspects of the module.

In the exemplary embodiment, the user can then turn the speech recognition engine back on for the user's session, by entering a designated DTMF tone that the processor is programmed to recognize. Alternatively, when the speech recognition engine is off, the processor can be programmed to recognize only the word "Unmute" as a command to cause the processor to turn back on the SR engine. For instance, the word "Unmute" can be a system-level grammar. Further, when the SR engine is off, it can still function to await the word "Unmute."

With this enhancement, a user can thus disable the speech recognition engine whenever the user wants to do so. For instance, if a user is listening to a long news story and is standing in a noisy airport, the user can say "Mute" to prevent a "barge-in" error such as where the background noise is erroneously recognized by the speech recognition engine. Advantageously, the word "unmute" is uncommon enough that it is not likely to be spoken unless a user intends to turn back on the SR engine.

#### **iv. Centrally Maintaining Personal Bookmarks**

In accordance with the exemplary embodiment, as noted above, user profile store 52 can further include, for each user, a set of one or more bookmarks that function as user-specific shortcuts to navigation points, such as URIs or menu items. Advantageously, by maintaining these bookmarks centrally on the voice command platform (or otherwise in a manner accessible by the platform), a user can make use of the bookmarks when the user calls the platform from any location.

Preferably, the enhanced system logic provides a mechanism for maintenance of these bookmarks, such as recording the bookmarks, receiving user requests to use the bookmarks, and applying the bookmarks to navigate to a designated navigation point. In the exemplary embodiment, this can be accomplished through a system-level application (e.g., a VXML application) that a user can access by speaking a system-level grammar such as "Bookmark." I.e., platform can be arranged to execute a bookmark-management application in response to the system-level grammar "Bookmark."

An exemplary bookmark-management application can provide a user with options such as (i) Save the current URI, (ii) Recall a bookmark, and (iii) Edit a bookmark. In response to user selection of the Save option, the bookmark-management application may direct the processor to save in the user's bookmark list the URI of the VXML document that the user was accessing before the user called the bookmark-management application (or the label of whatever other navigation point the user was accessing at the time). Further, the bookmark-management application may direct the processor to prompt the user for a shortcut name (i.e., an alias) for the bookmark, possibly providing the user with a suggested default name. Once the user speaks the name or agrees to use the default, the processor may store the name in the user's bookmark list together with the URI.

In response to user selection of the Recall option, the bookmark-management application may direct the processor to prompt the user to speak the name of a bookmark on the user's list. Once the user speaks the name, the processor may refer to the bookmark list to determine the corresponding URI and may then load the document from that URI and execute the document.

As a further enhancement, the bookmark-management application may direct the processor to recognize the command "Recall" followed by the name of the bookmark that the

user wants to recall, and to responsively load and execute the corresponding URI. For instance, in response to a user speaking "Recall Shopping," the bookmark-management application may direct the processor to refer to the user's bookmark list to find a URI that corresponds to the name "Shopping." The processor may then find the corresponding URI and load and execute the URI. (Note that the "Recall URI" grammar structure can itself be made a system-level grammar of the voice command platform, so that a user can call up a stored URI at any time, without first expressly calling up the bookmark-application.)

In response to user selection of the Edit option, the bookmark-management application may direct the processor to prompt the user to speak the name of a bookmark on the user's list. The processor may then prompt the user to indicate whether the user wants to rename or delete the entry. Other examples are possible as well.

#### **v. Applying User-Specific Designation of Speech Persona**

According to the exemplary embodiment, as noted above, user profile store 52 can indicate for each user a user-specific persona. A user-specific persona defines a voice façade by which the platform will present itself to the user. As such, the persona could define a specific TTS engine (of TTS engines 44) and/or voice prompt store (of voice prompt stores 46) that the platform should apply when it speaks to the user. Other aspects of a user-specific speech persona are possible as well. For instance, a given user's speech persona may dictate a particular tone and/or pitch for the TTS engine to use when generating analog speech signals.

The persona is user-specific, in that the platform could use one persona when interacting with one user and a different persona when interacting with another user. In the exemplary embodiment, the user profile record for a given user can include a table or other indication that

identifies a TTS engine and/or voice prompt set to use during sessions with the user and that may further indicate other parameters about the way that the speech should be presented to the user. Further, the persona designation can define more complex persona-selection logic, such as that the platform should use one persona when interacting with the user at one time (e.g., daytime) and a different persona when interacting with the user at another time (e.g., evening).

The persona designation for a given user may point specifically to a TTS engine and/or voice prompt set, or it may point more generally to a persona that the platform should use. For example, the persona designation may indicate the name of a particular TTS engine (e.g., as "English\_tts", "Spanish\_tts", "male\_tts", "female\_tts", etc.) and/or the label of a particular voice prompt directory (e.g., "Celebrity\_1", "Celebrity\_2", "male", "female"). Alternatively, the persona designation may indicate more generally a persona category, in which case the processor may refer to a stored translation table that indicates which TTS engine and/or voice prompt set to use so as to achieve the designated persona.

Further, the voice of the prompt set and the voice of the TTS engine can be made to approximate or match each other (e.g., both can be English speaking male voices of roughly the same pitch), so that a user profile record can indicate a TTS engine and the platform can select a corresponding voice prompt set (and vice versa). To do so, the platform can be programmed with system-level logic (e.g., a cross-reference table) that matches voice prompt sets with TTS engines and that indicates parameters such as pitch.

Thus, when a user calls the platform, processor 22 can programmatically consult user profile store 44 to determine which persona is to be used for speaking to the user. As the processor then executes an application for the user, the processor may then apply the designated TTS engine and/or voice prompt set. Further, the platform can have a default persona (e.g., a

standard "male" persona), which the processor can be programmed to use when the user's profile does not indicate that the platform should use another persona.

In the exemplary embodiment, a platform can also allow a user to change the user's persona selection mid-session. For instance, by speaking a system-level grammar such as "Change Persona", the user could direct the platform to execute a persona-management application. The persona-management application can tell the user which persona is currently set for the user and can prompt the user to select a different persona. In response to user-selection of a different persona, the processor can update the user's profile record accordingly and can begin to use the different persona.

Advantageously, a user-specific speech persona can be tailored to separately suit each particular user. For instance, a female user might prefer a female voice TTS engine and a corresponding female voice prompt set, while a male user might prefer a male voice TTS engine and corresponding mail voice prompt set. Other examples are possible as well.

**b. Enhancing Application Control  
Over the Voice Command Platform**

According to the exemplary embodiment, the enhanced system logic also provides for enhancing application control (or content-provider control) over the voice command platform. In one respect, this enhanced control can be provided by supplying the platform with one or more application-specific phoneme dictionaries that the platform can use to perform voice translation when executing a respective application. In another respect, the enhanced control can be provided by enabling an application to specify which of multiple TTS engines, voice prompt stores and/or phoneme dictionaries the platform should use when communicating with a user.

Each of these enhanced control functions will now be described.

i. **Using an Application-Specific Phoneme Dictionary**

As noted above, the system layer traditionally includes or has access to a core phoneme dictionary 42, which defines phonemes that the voice translator 24 can use in performing speech recognition. This dictionary thus enables the voice command system to recognize words spoken by a user.

A problem with this arrangement, however, is that the core dictionary might not define all of the possible utterances that users may make in connection with applications executed by the platform. For instance, a mapping application might include as grammars a list of street names and city names, many of which might have unconventional pronunciations. In order for the voice command platform to be able to interface with a user, the platform should ideally be able to understand utterances that are specific pronunciations unique to the content provider or application. At the same time, a platform provider should not have to overhaul its core dictionary in order to be compatible with all possible applications.

To overcome this problem, as noted above, then enhanced system logic can include one or more secondary phoneme dictionaries 48, each of which can be associated with a particular application (e.g., a particular URI or other navigation point) and/or a particular content provider. In the exemplary embodiment, each secondary phoneme dictionary can be a uniquely named data file that defines additions to the core dictionary, such as additional phonemes that the processor should recognize when executing the SR engine 36. For instance, the secondary dictionary associated with one application might be named "Phoneme\_46", while the secondary dictionary associated with another application might be named "Phoneme\_47".

Thus, when the processor executes an application, the processor can determine whether the application has an associated secondary dictionary 48. If so, the processor can apply that secondary dictionary 48 in conjunction with the core dictionary 42 as the processor executes the SR engine 36.

To associate each secondary phoneme dictionary with an application, the voice command platform can include or have access to a translation table or other logic that indicates, for a given application, which secondary phoneme dictionary to use. The table or other logic can be set up to correlate a given application with a given secondary phoneme dictionary and/or to correlate all applications by a given content-provider with a given secondary phoneme dictionary. Further, the table can correlate more than one secondary dictionary with a given application. Thus, to determine whether an application has an associated secondary phoneme dictionary, the processor can refer to the translation table or other logic.

The voice command platform can obtain secondary phoneme dictionaries in various ways. In the exemplary embodiment, for instance, a content-provider or other entity can provide the voice command platform with a secondary phoneme dictionary to be used for a given application (or applications).

For example, the content provider can provide the secondary phoneme dictionary in advance, i.e., before the platform loads and executes the application. Alternatively, an application can include a secondary phoneme dictionary as a component that the platform will load when the platform loads the application. (This can be similar to how a browser loads a graphic or other component when it loads a conventional HTML web page.) The enhanced system logic will then cause the processor to load the secondary dictionary, store the secondary dictionary for use during execution of the application, and use the secondary dictionary. In the

latter scenario, the processor can be programmed to delete the secondary dictionary when the processor finishes executing the application, or the processor can maintain the secondary dictionary for use again next time the application is executed.

**ii. Applying a Voice Translation Mechanism Designated by an Application**

In the exemplary embodiment, the enhanced system logic also enables an application to specify which of various voice translation mechanisms the platform should use during execution of the application. For instance, the logic can enable an application to specify which of multiple TTS engines 44, voice prompt stores 46 and/or secondary phoneme dictionaries 48 to use during execution of the application.

This facilitate this, each voice translation mechanism can have a unique identifier, and the processor can be programmed to recognize and respond to an instruction (e.g., a predefined command, tag, etc.) in an application that specifies the use of a particular voice translation mechanism. Each such instruction may, for instance, indicate a voice translation type and a voice translation value.

For example, as noted above, exemplary TTS engines might be named "English\_tts", "Spanish\_tts", "male\_tts", "female\_tts", etc., exemplary voice prompt stores might be named "Celebrity\_1", "Celebrity\_2", "male", "female", etc., and exemplary phoneme dictionaries can be named "Phoneme\_46" and "Phoneme\_47". Exemplary VXML tags to specify a voice translation mechanism might then take the following form:

```
<TTS = "Spanish_tts">  
<VOICEPROMPT = "Celebrity_1">  
<PHONEME = "Phoneme_46">
```

An application can be written to include one or more of these tags. And the VXML interpreter 26 can then be written to recognize these tags and to direct the processor to apply the designated voice translation mechanism.

Further, the platform can include or have access to a translation table or other logic that correlates unique voice translation keys (e.g., code numbers) with designated voice translation mechanisms. According to that table, a given voice translation key could be correlated with a specific TTS engine 44, a specific voice prompt store 46 and/or a specific secondary phoneme dictionary 48. For instance, voice translation key "3752" can be correlated with a combination of the "male\_tts" TTS engine and the "male" voice prompt store.

A suitable instruction in an application can then specify a particular voice translation key, such as by a VXML tag like the following:

```
<VOICETRANS = "3752">
```

In turn, the VXML interpreter can be written to recognize such a tag and to direct the processor to apply the corresponding voice translation mechanism(s).

With the exemplary arrangement, a content-provider can thus write a voice command application to take advantage of the ability to dynamically select a voice translation mechanism that the platform should use. For instance, a given application may include a voice translation tag at the start of its root document, which will direct the platform to apply a particular TTS engine, a particular voice prompt store and/or a particular secondary phoneme dictionary during execution of the application.

Further, by including a voice translation tag in an application, the application can cause the platform to switch between voice translation mechanisms dynamically during execution of

the application. For instance, the application may cause the platform to switch from one TTS engine to another during execution of the application.

In the exemplary embodiment, the voice command platform will preferably be programmed to treat particular voice translation mechanisms as default mechanisms, in the event no other designation is made. For example, the default setting may be to use the "English\_tts" TTS engine, the "female" voice prompt store, and no secondary phoneme dictionary. Other examples are possible as well.

Finally, another way for an application to specify a voice translation mechanism to use is for the application or content-provider to provide the platform with the voice translation mechanism. As noted above, for example, the application or content-provider can provide the platform with a secondary phoneme dictionary 48. Further, since the TTS engines 44 and voice prompt stores 46 are also preferably software-based, it would be possible for an application or content-provider to similarly provide the platform with an application-specific TTS engine and/or voice prompt store to use during execution of the application.

### **c. Monitoring and Use of Session Information**

According to another aspect of the exemplary embodiment, the enhanced system logic also provides for maintaining and using information concerning user sessions with the voice command platform. In one respect, this functionality thus involves maintaining user session information. And in another respect, this functionality then involves using the session information to enhance user interaction with the platform in some manner. By way of example, two enhancements that a user's session information can facilitate are (1) automatically transitioning a user to expert mode based on the user's navigation history, and (2) automatically restoring a session that was abruptly cut off by a system disconnect or other event.

The process of maintaining user session information and using the information to perform these exemplary functions will now be described.

**i. Maintaining User Session Information**

As noted above, user profile store 52 can include a record of user session information, such as current session state information as well as user navigation history. In the exemplary embodiment, the processor 22 is preferably programmed to maintain this session information for each user, i.e., to keep track of each user's session with the platform.

In terms current session state information, for instance, the processor may be programmed to record in the user profile an indication of whether or not the speech recognition engine 36 is muted for a given user's session, as well as an indication of which TTS engine and/or voice prompt set is currently active for the user's session. These parameters can be set as flags or other parameters in the user's profile record.

In terms of navigation history, the processor may be programmed to maintain a list of the URIs accessed by the user, beginning with the most recent. In addition, as noted above, the processor may maintain a summary table that indicates, for each URI, how many times the user has accessed the URI. The processor may thus increment an entry in the summary table each time a user accesses the same URI.

Further, the processor may also be programmed to maintain in the user profile a record of how many times each user has contacted the voice command platform, possibly together with indications of the dates and/or times that the user has contacted the platform.

ii. **Automatically Transitioning a User to  
Expert-Mode, Based on Navigation History**

Generally speaking, a voice command application can define a number of voice prompts, acceptable response grammars, and associated logic that the platform is to execute in response to particular responses or other events. Thus, as a user interacts with the voice command application, the platform would present designated voice prompts to the user, receive grammars from the user, and carry out logic in response to the grammars or other events.

A user who interacts regularly with the same application, however, may tire of listening to the same voice prompts over and over, and having to regularly provide the same response grammars to each voice prompt. To enhance the user experience, a robust voice command application can therefore define multiple sets of voice prompts, acceptable response grammars and/or associated logic.

One set might be considered a standard set, which would be appropriate for interaction with the typical user. And another set might be considered an expert set, which would be appropriate for interaction with a user considered to be an expert. Thus, depending on whether the user who is interacting with the voice command application is designated as an expert user or not, the application may then instruct the processor to apply either the standard set or the expert set.

An expert set of voice prompts and/or acceptable response grammars may be more streamlined than the standard set, such that a user can more quickly navigate. For instance, when a given application is going to prompt a user to select between a number of choices, a standard voice prompt might be a full statement of the list of choices followed by a closing voice prompt such as "Please state your response now." In contrast, an expert voice prompt at the same

navigation point might be a scaled down version of the list of choices and/or a shorter closing voice prompt such as "Ready." Alternatively, the expert voice prompt might omit portions of the standard voice prompt in their entirety and might substitute a short tone or other mechanism to solicit a response.

As a specific example, at a given navigation point in a weather reporting application, the weather reporting application might define as a standard voice prompt the words, "If you would like to hear today's weather, please say 'today.' If you would like to hear an extended forecast, please say 'extended.' Please speak your response now." And at the same navigation point, the weather reporting application might define an expert voice prompt, "Today or Extended?" followed by a quick prompting tone. Depending on whether the user who is currently interacting with the application is designated as an expert or not, the application may then instruct the platform to apply either the standard prompt or the expert prompt.

In accordance with the exemplary embodiment, a given user can be designated as either a standard-mode user or an expert-mode user. (Other such designations or use-levels are also possible. For instance, a user could be a beginner, intermediate or advanced user.) Further, for a given user, different use-level designations can be specified for different navigation points. For instance, a user may be designated as a standard-mode user for a particular application (e.g., a particular URI), and the user may be designated as an expert-mode user for another application or for a particular navigation point within an application.

Preferably, the user profile store 52 will include an indication, per user, of a use-level, such as whether or not the user is an expert-mode user. If a user is globally designated as an expert mode user for all applications that the voice command platform executes, the user's profile record may include a flag that indicates the user is an expert-mode user. Alternatively, if the

user is designated as an expert-mode user for some navigation points and a standard-mode user for others, then the user's profile record can include a table that indicates, per navigation-point, whether the user is an expert-mode user. The table may, for instance, list URIs (and/or other navigation point labels) and, for each URI, provide a flag indicating whether the user is an expert-mode user.

At a given navigation point in a voice command application, the application can then include logic that instructs the processor to apply a standard-mode prompt if the user is a standard-mode user or to apply an expert-mode prompt if the user is an expert-mode user. In response to that logic, the processor may consult the user profile store to determine whether or not the user is an expert-mode user. If the user is an expert-mode user, the processor may then apply the expert-mode prompt. And if the user is a standard-mode user, the processor may apply the standard-mode prompt.

In accordance with the exemplary embodiment, as noted above, the enhanced system logic further includes a mechanism to automatically switch a user from standard mode to expert mode, either generally or for specific navigation points. The transition to expert mode can occur based on any designated event or combination of events.

For example, as noted above, the processor can be programmed to review a user's navigation history and to automatically transition a user to expert mode after the user has accessed a particular navigation point, particular navigation points, and/or the voice command platform generally, more than a threshold number of times. The processor may perform this review each time a user calls the voice command platform, or at other designated times.

This logic can be as simple or as complex as desired. For instance, the logic could direct the processor to switch a user to expert mode for a given navigation point after the user has

accessed the navigation point at least 10 times. Alternatively, the logic could direct the processor to switch a user to expert mode for a given application once the user has accessed the application at least 20 times and has accessed at least two second level navigation points in the application. Other examples are possible as well.

In the exemplary embodiment, when the processor determines that the user should be transitioned to expert mode, the processor could prompt the user with an announcement indicating the benefits of the transition and requesting the user's consent to the transition. Alternatively, the processor could perform the transition without asking the user for consent. Further, the processor could perform the transition even when the user is offline, such as at nighttime when the platform might have more available processing power.

### **iii. Using Session Information to Help Restore a Session After System Disconnect**

As noted above, the enhanced system logic can also facilitate restoring a user to where the user left off in a voice command session, when a session ends abruptly. This can occur, for instance, if the user's telephone connection to the platform gets cut off (e.g., as a result of a coverage drop, signal fade, hang-up or out-dialed call), if the system encounters an error that causes the session to be dropped, or in other instances.

In the exemplary embodiment, when a system disconnect occurs, the processor will preferably maintain an indication of the disconnect for a period of time (e.g., 15 minutes) in user profile store 52, together with an indication of the user's navigation points and perhaps other session information at the moment of the disconnect. Provided that the platform keeps a log of the navigation points that the user has accessed, as described above, the user's latest navigation point could be the most recent navigation point in that log.

When a user calls the platform, the processor may be programmed to consult the user's profile record to determine whether a system disconnect occurred during a session with the user. If the processor determines that a system disconnect occurred, the processor may then prompt the user to indicate whether the user wants to resume where the user left off or, rather, whether the user wants to begin from the start. After the user responds, the processor then either loads and executes an application where the user left off or begins a voice command session normally.

For example, during a session with the voice command platform, a user may be navigating through a set of driving directions. Preferably, the content provider has arranged each step in the driving directions at a separate URI (where one URI leads to the next). When the user navigates to step 5 of the driving directions, the processor may record the URI of step 5 in the user's navigation history, preferably together with a name of the navigation point (e.g., "Step 5 of driving directions"). At that point, a system disconnect might occur. The processor may then record a flag or other indication in the user profile, indicating that a system disconnect occurred, and noting the time of the disconnect.

When the user then calls back the platform, the processor may consult the user's profile and determine that a system disconnect occurred. In response, the processor may programmatically prompt the user to indicate whether the user wants to resume where the user left off. For instance, based on the user's navigation history, the processor may identify the navigation point that the user most recently visited and may then prompt the user accordingly, such as "Say 'Step 5' to resume Step 5 of the Driving Directions or say 'Voice Command' to start over." Once the user responds, the processor may then either load the associated URI.

**d. Providing Unified Message Notification**

According to the exemplary embodiment, user profile store 52 can include a consolidated summary of the number of messages waiting for the user at multiple separate message portals, and the processor can be programmed to provide a consolidated or unified message notification to the user. This function can thus involve (i) maintaining consolidated message waiting information for a user and (ii) providing the consolidated notification to the user.

In order to maintain consolidated message waiting information, the platform provider will preferably receive update messages from various message portals (e.g., an e-mail portal, a voice mail portal, a fax portal, etc.), indicating the number of messages waiting for the user at each message portal. To accomplish this, a user may register with various message portals to authorize the message portal to report message-waiting counts to the voice command platform (or to an intermediate entity that functions to store and forward the information to the voice command platform). The various message portals will then send update messages to the voice command platform.

Preferably, the update messages that each portal sends to the voice command platform will be in a predetermined standard format that processor 20 is programmed to recognize. For instance, each update message could be an IP-based message that includes an XML indication of (i) message portal name, (ii) user ID (e.g., the user's message portal ID or, more preferably, the user's voice command platform ID), and (iii) priority. As an example, a message from a Yahoo e-mail portal might specify that the user has 4 messages of urgent priority waiting at Yahoo, while a message from a corporate e-mail portal might specify that the user has 2 messages of standard priority waiting at the corporate e-mail system.

Each portal can be programmed to send such an update message to the voice command platform each time the number of messages waiting for the user at the portal changes. For instance, once a user reads or hears a message that was waiting at a given message portal, the message portal may decrement the number of messages-waiting by one and send an update message to the voice command platform.

In the exemplary embodiment, processor 20 is then programmed to receive the update message count per user from each of multiple message portals and to responsively update the user's profile record with a current summary of the number of messages waiting at the various portals. For instance, the user profile record might include a table that lists various message portals, the type of each portal, and the number of messages (of various priority levels) waiting for the user at the given portal. The processor can thus readily update the table in response to each new update message that it receives from a message portal.

When a user calls the platform, and/or at another designated time during a session with the platform, the processor preferably provides the user with a consolidated message notification, based on the summary maintained in the user profile. The consolidated message notification will indicate to the user how many messages are waiting for the user to retrieve from at least two different message portals. For instance, a system-level root VXML application can welcome the user to the platform and can then announce to the user the number of messages waiting for the user at the various portals. The system-level application may carry out this function before or after providing the user with an initial Ready prompt.

The user may then opt to browse to the designated message portals so as to have the messages read to the user. For instance, if the user has messages waiting in a Yahoo e-mail

system, the user may command the platform to load a Yahoo e-mail application, which may function to read the user's e-mail messages to the user.

### **3. Provisioning**

In the exemplary embodiment, a provisioning system may be established so as to allow users to set up and manage their profiles. For instance, a provisioning server may be provided on the Internet and may function as an intermediary between the voice command platform and users. As such, the provisioning server may have real-time access to the user profile store 52 and may function to validate data (e.g., user-specified changes to profile records) before writing them to the profile store.

A user may connect with the provisioning server from a web browser on a client computer. Once connected, the user would log in (e.g., by providing a user ID, which the provisioning server would match with a profile record), and the provisioning server would then present the user with a consolidated summary of the user's profile record. The consolidated summary may include information such as: (i) the persona and/or default TTS engine designated for the user, (ii) a list of the user's personal-grammars and their corresponding actual grammars, (iii) the user's expert mode state, (iv) the user's navigation history, (v) a list of the user's bookmarks and (vi) the user's consolidated messaging summary. The provisioning server may then permit the user to modify some or all of this information via the web interface.

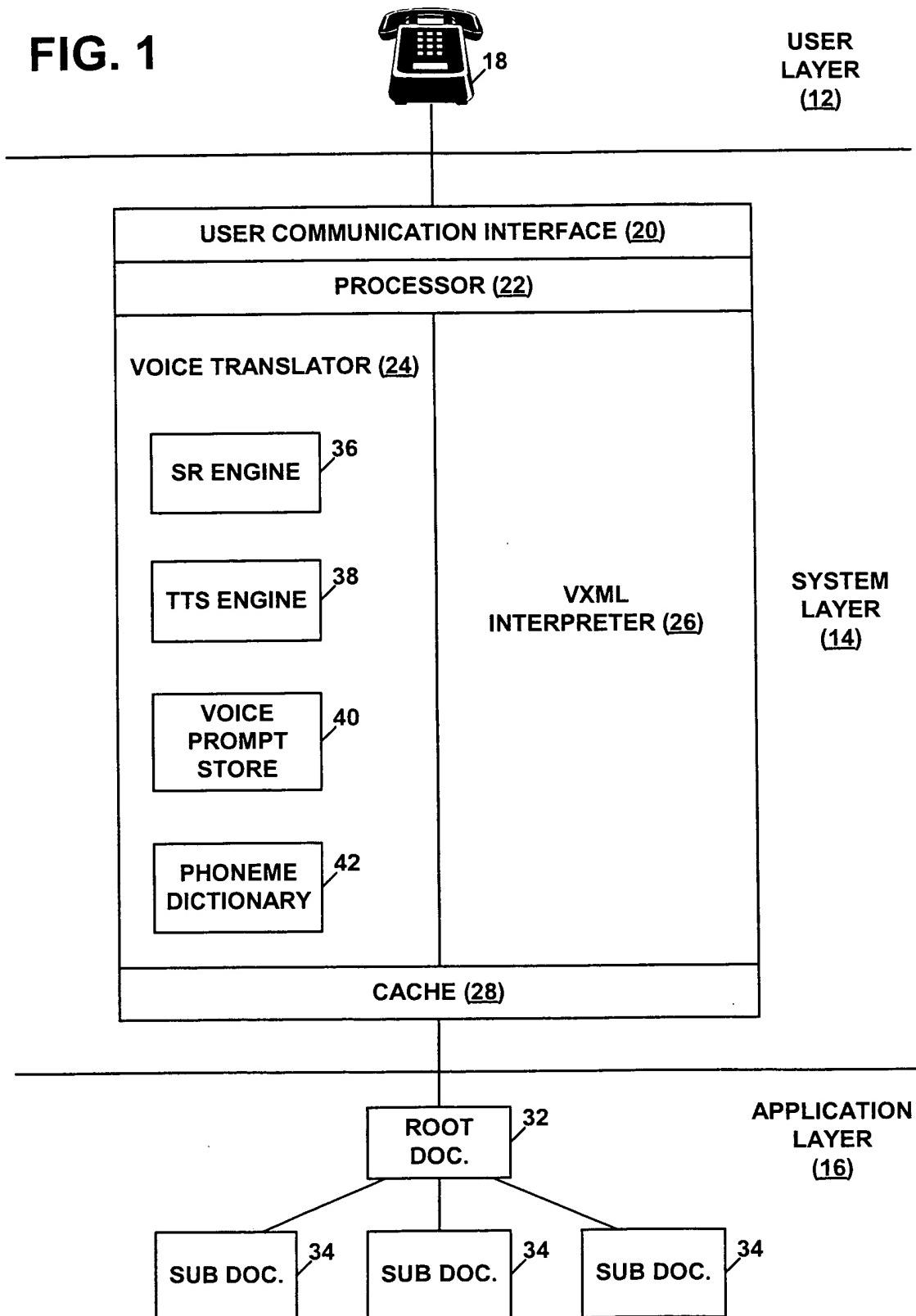
In addition, the provisioning server may function as an interface through which content providers can provide the voice command platform with secondary phoneme dictionaries. Preferably, a content provider would log in, and the provisioning server would then allow the content provider to upload or modify secondary phoneme dictionaries and associations with

applications. The provisioning server may then store the secondary dictionaries on the voice command platform or elsewhere to be accessed and used by the platform.

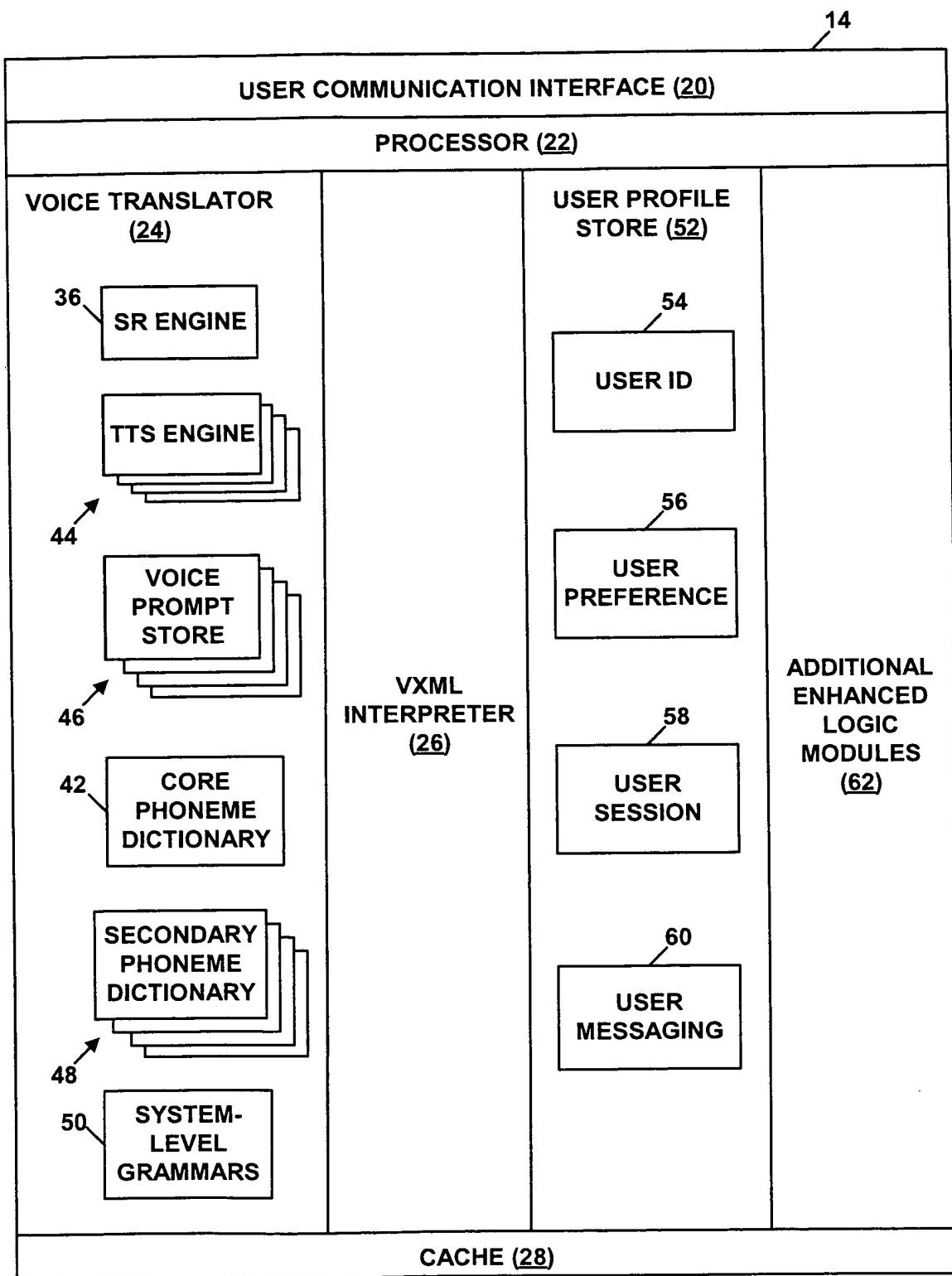
#### **4. Conclusion**

Exemplary embodiments of the present invention have been described above. Those skilled in the art will understand, however, that changes and modifications may be made to these embodiments without departing from the true scope and spirit of the invention, which is defined by the claims.

**FIG. 1**



# FIG. 2



**Cisek, Kelly**

---

**From:** Aaronson, Larry  
**Sent:** Tuesday, August 07, 2001 3:18 PM  
**To:** Steven Funk (E-mail); Kevin Packingham (E-mail)  
**Cc:** Tim Nehls (E-mail)  
**Subject:** Patent Matter -- Sprint Docket Nos. 1700, 1730, 1731 and 1732

**CONFIDENTIAL & PRIVILEGED**  
**ATTORNEY-CLIENT PRIVILEGE**

Steve and Kevin,

Attached for your review and comment are four sets of patent claims directed to the Voice Command system improvements. I have categorized these sets of claims under Sprint Docket Nos. 1700, 1730, 1731 and 1732. However, as we have discussed, the division among these categories is different than what we had initially contemplated. We will therefore need to re-address the inventorship issue, to identify who the inventors are for each group of claims. (I suggest that we don't make that determination yet; rather, let's wait until we have finalized the claims and finalized the division of the claims into separate patent applications).

Steve, you and I should discuss the coverage of these claim sets at your earliest convenience.

Kevin, please review these claims carefully and let me know if you feel that the claims recite what you believe to be the subject matter of your (and your co-inventors') inventions. In this regard, please bear in mind that the claims will be the legal statements that define the boundary of your invention. The claims are what a person would infringe or not infringe. I have therefore tried to claim your inventions broadly enough to provide commercial value for Sprint, but not so broadly as to describe what might already exist.

Once we finalize the claim sets, I will also need to write summary sections for each of the respective patent applications, and I may need to revise the common detailed description section to some extent, for consistency with the claims.

Finally, I'd like to remind you again of the duty of disclosure imposed by the U.S. Patent Laws. If you or any of the other inventors are aware of any prior art that might be considered to bear on the patentability of this invention, please let me know as soon as possible, so that we can be sure to submit that prior art to the Patent Office. Failure to comply with the duty of disclosure can render a resulting patent unenforceable in federal court.

Please let me know if you have any questions.

Thanks.

Larry H. Aaronson  
McDonnell Boehnen Hulbert & Berghoff  
300 South Wacker Drive  
Chicago, Illinois 60606  
312-913-0001 phone  
312-913-2141 direct  
312-913-0002 fax



VC



VC



VC



VC

ments -- CLAIMments -- CLAIMments -- CLAIMments -- CLAIM

**SPRINT CORPORATION**  
Patent Matter – Sprint Docket No. 1731  
"Method and System for Use of Navigation History  
in a Voice Command Platform"

**Draft Abstract and Patent Claims**

*Confidential & Privileged  
Attorney-Client Privilege*

**TOPICS COVERED:**

- Restore from System Disconnect
- Expert Transition

**ABSTRACT:**

A method and system for using navigation history in a voice command platform. The voice command platform maintains, for each user, a navigation history record indicating which voice command navigation points the user has accessed during one or more sessions with the platform. The platform may then use the navigation history as a basis to restore a voice command session with the user after a system disconnect and/or to determine that the user should be automatically transitioned to an expert-user mode.

**CLAIMS:**

1. A voice command platform comprising:  
a user communication interface for communicating with users via a telecommunications network;  
a processor;  
an application-processing module executable by the processor to process voice command applications, the voice command applications having navigation points, and the voice command applications defining user-prompts, allowed grammars and application-logic, wherein the processor processes voice command applications during voice command sessions with users;  
a user profile store including a navigation history record respectively for each of a plurality of users, the navigation history record for a given user identifying navigation points of voice command applications that the processor has processed during at least one voice command session with the given user.
2. The voice command platform of claim 1, further comprising:  
navigation-recording logic executable by the processor to record in the navigation history record for the given user an indication of a navigation point of a voice command application that the processor has processed during a voice command session with the user.
3. The voice command platform of claim 1, wherein the navigation-recording logic is executable by the processor to record in the navigation history record for the given user each navigation point accessed during the voice command session with the user.
4. The voice command platform of claim 1, further comprising:  
session-restore logic executable by the processor to restore a given voice command session with the given user based on the navigation history record for the given user.
5. The voice command platform of claim 4, wherein the session-restore logic is executable by the processor to determine that a system disconnect occurred during the given voice command session, and to thereafter restore the given voice command session.

6. The voice command platform of claim 5, wherein the session-restore logic is further executable by the processor to prompt the user for consent to restore the given voice command session.

7. The voice command platform of claim 5, wherein:

the user profile store includes an indication for the given user indicating that the system disconnect occurred; and

the session-restore logic is executable by the processor to determine, based on the indication, that the system disconnect occurred.

8. The voice command platform of claim 5, wherein the session-restore logic is executable to restore the given voice command session by a process comprising:

using the navigation history record for the given user to identify a voice command application that the processor was processing at the time the system disconnect occurred; and

loading and executing the voice command application.

9. The voice command platform of claim 5, wherein the navigation history lists navigation points in order of navigation.

10. The voice command platform of claim 4, wherein the session-restore logic is executable to restore the given voice command session for a period of approximately 15 minutes after a system disconnect of the given voice command session.

11. The voice command platform of claim 4, wherein each of a plurality of the voice command applications are VXML applications, and each of a plurality of the navigation points are Universal Resource Indicators.

12. The voice command platform of claim 1, further comprising:

expert-mode-transition logic executable by the processor to automatically transition the given user to expert-mode user status based on the navigation history record for the given user.

13. The voice command platform of claim 12, wherein the expert-mode-transition logic is executable:

to make a determination, based on the navigation history record for the given user, that the given user has accessed a navigation point at least a threshold number of times; and

to set an expert-mode user flag in a profile record for the user, in response to the determination.

14. In a voice command platform of the type that communicates with users via a telecommunications network and that executes voice command applications during voice command sessions with users, a method comprising:

storing, respectively for each of a plurality of users, a navigation history log indicating navigation points of voice command applications that the platform has executed during at least one voice command session with the user.

15. The method of claim 14, further comprising:

using the navigation history log to restore a previous voice command session with the user.

16. The method of claim 15, wherein using the navigation history log to restore a previous voice command session with the user comprises:

determining that a system disconnect occurred from the previous voice command session;

identifying, based on the navigation history log, a given navigation point of a given voice command application that the platform was executing at the time the system disconnect occurred;

loading the given voice command application from the given navigation point; and  
executing the given voice command application.

17. The method of claim 16, further comprising restoring the previous voice command session with the user at the initiation of a subsequent voice command session with the user.

18. The method of claim 17, further comprising prompting the user for consent to restore the previous voice command session.

19. The method of claim 14, further comprising:  
using the navigation history log to determine that the user should be automatically transitioned to expert-mode user status; and  
automatically transitioning the user to expert-mode user status.

20. The method of claim 19, wherein:  
using the navigation history log to determine that the user should be automatically transitioned to expert-mode user status comprises using the navigation history log to determine that the user should be automatically transitioned to expert-mode user status with respect to a given navigation point; and  
automatically transitioning the user to expert-mode user status comprises automatically transitioning the user to expert-mode user status with respect to the given navigation point.

21. The method of claim 19, wherein using the navigation history log to determine that the user should be automatically transitioned to expert-mode user status comprises:  
determining, based on the navigation history log, that a given navigation point has been accessed at least a threshold number of times during at least one voice command session with the user; and  
responsively determining that the user should be automatically transitioned to expert-mode user status with respect to at least the given navigation point.

22. A voice command platform comprising:

a processor;

a stored indication, for each of a plurality of users, of a use-level of the user and a navigation history of the user, the use-level being selected from the group consisting of (i) expert-mode and (ii) not-expert-mode; and

logic executable by the processor to switch the use-level of a user from not-expert-mode to expert-mode, in response to the navigation history of the user.

23. The voice command platform of claim 22, wherein the logic is further executable by the processor to prompt the user for authority to switch the user's use-level.

24. The voice command platform of claim 22, further comprising:

a voice command application including expert-mode logic and non-expert-mode logic, wherein the processor executes the expert-mode logic when the voice command platform is interacting with a user for whom the user profile store specifies an expert-mode use-level, and the processor executes the not-expert-mode logic when the voice command platform is interacting with a user for whom the user profile store specifies a not-expert-mode use-level.

25. In a voice browser system arranged to execute voice-tag applications and to interface between voice tag applications and users, a method comprising:

maintaining a navigation-history record that indicates a user's navigation history through at least one of the voice-tag applications via the voice browser system;

maintaining a use-mode record that indicates whether the user is an expert-user of the at least one voice-tag application;

automatically setting the use-mode record to indicate that the user is an expert-user of the at least one voice-tag application, in response to the navigation-history record; and

when executing the at least one voice-tag application, interfacing with the user according to the use-mode record.

26. The method of claim 25, wherein the at least one voice-tag application defines a standard set of logic including a standard set of voice prompts and the at least one voice-tag

application defines an expert set of logic including an expert set of voice prompts, and wherein interfacing with the given user according to the use-mode record comprises:

making a determination that the use-mode record indicates that the user is an expert-user of the at least one voice-tag application; and

responsive to the determination, executing the expert set of logic rather than the standard set of logic.

27. The method of claim 26, wherein voice prompts of the expert set are shorter in duration than voice prompts of the standard set.

28. The method of claim 26, wherein the standard set of voice prompts includes a voice prompt for a given menu item, and the expert set of voice prompts includes a tone prompt for the given menu item.

29. The method of claim 28, wherein automatically setting the use-mode record to indicate that the user is an expert-user of the at least one voice-tag application, in response to the navigation-history record, comprises:

determining that the user has accessed the given menu item at least a threshold number of times, and responsive setting the use-mode record to indicate that the user is an expert-user of the at least one voice-tag application.

30. The method of claim 25, wherein the at least one voice-tag application defines a standard prompt for a given menu item and an expert prompt for the given menu item, and wherein interfacing with the given user according to the use-mode record comprises:

making a determination that the use-mode record indicates that the user is an expert-user of the at least one voice-tag application; and

responsive to the determination, executing the expert prompt rather than the standard prompt.

31. The method of claim 30, wherein the expert prompt is shorter in duration than the standard prompt.

32. The method of claim 31, wherein the standard prompt is a voice prompt and the expert prompt is a tone prompt.

**From:** Aaronson, Larry  
**Sent:** Wednesday, September 05, 2001 8:31 AM  
**To:** 'Packingham, Kevin'  
**Cc:** Steven Funk (E-mail); Tim Nehls (E-mail)  
**Subject:** RE: Patent Matter -- Sprint Docket Nos. 1700, 1730, 1731 and 1732

3rd email

CONFIDENTIAL & PRIVILEGED  
ATTORNEY-CLIENT PRIVILEGE

Kevin,

Thanks. I am sending out the final patent applications, together with formal papers (for each application, (i) a Declaration and Power of Attorney and (ii) an Assignment) to you overnight tonight. In the cover letter accompanying the applications, I will ask that the inventor(s) for each application review the application and the formal papers and then sign the formal papers (and have the signatures on the Assignments witnessed).

Note that the "Background", "Brief Description of the Drawings", and "Detailed Description of Exemplary Embodiments" sections of the four applications are identical. So each inventor can review those sections in just one of the applications. However, the other sections of each application differ, so each inventor will need to review those other sections per application before signing the respective Declaration and Power of Attorney.

I look forward to receiving the signed papers from you, so that we can file these four patent applications with the U.S. Patent and Trademark Office.

Also, I again remind you of the duty of disclosure imposed by the U.S. Patent Laws. If you are aware of any information about existing or proposed systems that would include features in any way similar or analogous to those that we have claimed in these applications, you should let me know as soon as possible, so that I can inform the Patent Office if appropriate. Please pass this advice on to the other inventors as well, since each inventor is bound by the duty.

If you have any questions, please contact me.

Thanks.

Larry H. Aaronson  
McDonnell Boehnen Hulbert & Berghoff  
300 South Wacker Drive  
Chicago, Illinois 60606  
312-913-0001 phone  
312-913-2141 direct  
312-913-0002 fax

-----Original Message-----

From: Packingham, Kevin [mailto:[KPacki01@sprintspctrum.com](mailto:KPacki01@sprintspctrum.com)]  
Sent: Monday, September 03, 2001 6:57 AM  
To: Aaronson, Larry  
Cc: Steven Funk (E-mail); Tim Nehls (E-mail)  
Subject: RE: Patent Matter -- Sprint Docket Nos. 1700, 1730, 1731 and 1732

2nd email

1700 - Kevin, Elizabeth, Terry  
1730 - Kevin, Elizabeth  
1731 - Kevin  
1732 - Kevin, Rob

Kevin Packingham

Sprint PCS Business Marketing  
Office: (913) 762-7575      PCS: (913) 485-6508

---

-----Original Message-----

From: Aaronson, Larry [mailto:Aaronson@mbhb.com]  
Sent: Tuesday, August 28, 2001 2:17 PM  
To: Kevin Packingham (E-mail)  
Cc: Steven Funk (E-mail); Tim Nehls (E-mail)  
Subject: Patent Matter -- Sprint Docket Nos. 1700, 1730, 1731 and 1732

1st email

CONFIDENTIAL & PRIVILEGED  
ATTORNEY-CLIENT PRIVILEGE

Kevin,

As a reminder, we should determine which people to list as inventors in the voice-command improvement patent applications that we are now finalizing. A person should be named as a joint inventor in a patent application if the person was jointly involved in making the invention -- i.e., if the person participated in conceiving of and/or figuring out how to practice any element recited by the patent claims in the patent application. With that in mind, please consider the each of the four separate groups of claims (corresponding to the four patent applications) that we discussed last week and let me know who we should designate as inventor(s) for each separate group.

Please let me know if you have any questions.

Thanks.

Larry H. Aaronson  
McDonnell Boehnen Hulbert & Berghoff  
300 South Wacker Drive  
Chicago, Illinois 60606  
312-913-0001 phone  
312-913-2141 direct  
312-913-0002 fax

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**